TAFE NSW

Digital Electronics - Part 1

tafensw.edu.au

Workbook

Author: Barry Duncan
This workbook is copyright protected and must not be reproduced without written consent (except under the *fair dealing* exceptions specified in the copyright act). For more information on what constitutes fair dealing visit: www.copyright.org.au

---

Digital Electronics - Part 1

Version 3.0 - October 2024

Printed for exclusive use by students of:
The Electronics Section
Ultimo Campus of TAFE NSW
Mary Anne St
Ultimo NSW 2007

Contents

Introduction _____ I-3

Textbooks _____ I-4

Lesson organiser _____ I-5

Required reading _____ R-1

Section 1 An introduction to digital electronics _____ 1-1

Skill Practice 1 _____ 1-23

Review questions _____ 1-37

Section 2 Logic gates _____ 2-1

Skill Practice 2 _____ 2-27

Review questions _____ 2-41

Section 3 Combinational logic _____ 3-1

Skill Practice 3 _____ 3-23

Review questions _____ 3-33

Section 4 Boolean algebra _____ 4-1

Skill Practice 4 _____ 4-21

Review questions _____ 4-31

Section 5 Logic families _____ 5-1

Skill Practice 5 _____ 5-25

Review questions _____ 5-31

Section 6 Number systems in digital electronics _____ 6-1

Skill Practice 6 _____ 6-27

Review questions _____ 6-33

Section 7 Codes in digital electronics _____ 7-1

Review questions _____ 7-17

Section 8 Digital displays _____ 8-1

Skill Practice 8 _____ 8-23

Review questions _____ 8-35

Section 9 Interfacing techniques 9-1
Skill Practice 9 9-23
Review questions 9-31
Section 10 RS flip flops 10-1
Skill Practice 10 10-23
Review questions 10-35
Appendices
Appendix 1: Digital test equipment
Appendix 2: Circuits that implement RS flip flops with synchronous R & S inputs
Appendix 3: Sample digital IC data (the 74ALS00)
Answers Section 1 A-1
Section 2 A-4
Section 3 A-7
Section 4 A-10
Section 5 A-13
Section 6 A-16
Section 7 A-18
Section 8 A-20
Section 9 A-23
Section 10 A-26

Introduction

This workbook is the first of two that have been written to help students satisfy the assessment requirements for the following Units:

- UEEEC0069 - Troubleshoot digital sub-systems

More information about electrotechnology training packages and the above Units can be found at www.training.gov.au/home/tga. Type the course/unit number in the "Quick search" and press return.

Textbooks

The following textbooks can be used to supplement the notes in this workbook. Most (if not all) of these books are available from the library in Building D.

- Bignell (1994). *Digital electronics* (3rd edition). Delmar Publishers.
- Bogart (1992). *Introduction to digital circuits*. McGraw Hill.
- Cook (1999). *A first course in digital electronics*. Prentice Hall.
- Edwards & Myer (1993). *Electronics; a basic course* (2nd edition). McGraw Hill.
- Floyd (2002). *Digital fundamentals* (7th edition). Prentice Hall.
- Gates (2001). *Introduction to electronics* (4th edition). Thompson Printing.
- Green (1993). *Digital electronics for technicians* (4th edition). Longman Scientific and Technical.
- *Kleitz (2002). *Digital electronics: A practical approach* (6th edition). Prentice Hall.
- Maxfield (2003). *Bebop to the Boolean boogie: an unconventional guide to electronics fundamentals, components and processes* (2nd edition). Newnes.
- Predko (2005). *Digital electronics demystified.* McGraw Hill.
- Tocci, Moss & Widmer (2004). *Digital systems: Principles and applications* (9th edition). Prentice Hall.
- Tokheim (2003). *Digital electronics: Principles and applications* (6th edition). McGraw Hill.
- Thompson (2001). *Digital electronics: A simplified approach*. Prentice Hall.

* Recommended

Lesson organiser

The following is a suggestion only. It may be changed to meet student needs and/or different modes of delivery.

Lesson	Section	Activity
1	Pt 1 - 1	An introduction to digital electronics Skill practice 1 - An introduction to testing and measuring digital ICs
2	Pt 1 - 2	Logic gates Skill practice 2 - Practise testing and measuring digital ICs
3	Pt 1 - 3	Combinational logic Skill practice 3 - An introduction to testing and measuring combinational logic circuits
4	Pt 1 - 4	Boolean algebra Skill practice 4 - Practise testing and measuring combinational logic digital circuits
5	Pt 1 - 5	Logic families Skill practice 5 - Practise testing and measuring mixed family combinational logic circuits
6	Pt 1 6 & 7	Number systems in digital electronics Codes in digital electronics Skill practice 6 - Practise testing and measuring digital ICs
7	Pt 1 - 8	Digital displays Skill practice 8 - An introduction to testing and measuring digital displays
8	Pt 1 - 9	Interfacing techniques Skill practice 9 - An introduction to testing and measuring digital interfacing circuits
9	Pt 1 - 10	RS flip flops Skill practice 10 - Practise testing and measuring sequential logic circuits (RS flip flops)
10	■ **Possible assessment - Refer to the UAG for details**	
	Pt 2 - 1	The D-type flip flop Skill practice 1 - Practise testing and measuring sequential logic circuits (D-type flip flops)

Lesson	Section	Activity
11	Pt 2 - 2 Pt 2 - 3	The J-K flip flop Skill practice 2 - Practise testing and measuring sequential logic circuits (J-K flip flops) Multivibrators Skill practice 3- Practise testing and measuring monostable multivibrator circuits
12	Pt 2 - 4	Counters Skill practice 4 - Practise testing and measuring sequential logic digital couner ICs
13	Pt 2 - 5	Shift registers Skill practice 5 - Practise testing and measuring sequential logic shift register ICs
14	Pt 2 - 6	Encoders and decoders Skill practice 6 - Practise testing and measuring digital decoder ICs
15	Pt 2 - 7	Multiplexers and demultiplexers Skill practice 7 - Practise testing and measuring digital multiplexer and demultiplexer ICs
16	Pt 2 - 8	Analog to digital and digital to analog converters Skill practice 8 - Practise testing and measuring ADCs and DACs
17	■ **Possible assessment - Refer to the UAG for details**	
18	■ **Possible assessment - Refer to the UAG for details**	

Note: The lesson notes for lessons 10 to 16 are in the workbook called *Digital Electronics - Part 2*. The lesson organiser for Part 2 is provided here to give you the full schedule of what is to be taught in this unit.

Required reading

Purpose These notes are provided to support your revision on aspects of this Unit that have been taught before in other Units (typically more than once) and will not be taught again but may be assessed.

This issue is explained in more detail on the following pages.

 Copyright: Not to be reproduced without permission

Introduction

As you probably know, *units of competence* that make up qualifications in the Electrotechnology Training Package (like CIII Electronics and Communications) prescribe what must be taught and assessed.

This is done using *elements of competence* together with *performance criteria* for each element. More detail on these is provided in the *assessment requirements* for the unit using two lists: one is the list of *performance evidence*; and the other is the list of *knowledge evidence*.

There are some performance criteria (together with performance evidence and/or knowledge evidence in the assessment requirements) that appear in many units and these include items on the following topics:

- work health and safety (WHS)
- using sustainable energy practices
- the appropriate use and completion of documentation
- appropriate communication with supervisors, customers and other stakeholders.

It makes sense that these performance criteria are common to multiple units because, taking WHS as an example, it's not reasonable to claim competence when the work was completed successfully but in an unsafe way.

As well as the topics listed above, there are performance criteria and/or assessment requirements related to electrical/electronic theory and practice that appear in two or more units. This occurs when the people who have developed the training package (not TAFE NSW) have decided that it's appropriate that they should do so. An example of this for UEEEC0069 is the requirement for you to demonstrate your knowledge of how to connect a digital multimeter to a circuit to measure potential difference which you have previously learnt about when undertaking UEECD0043 (formerly UEENEEE104A).

Importantly, where all these topics have been comprehensively taught before, it is usually unnecessary to teach them a second or third time. That said, we are still expected to ask you to provide knowledge evidence and/or performance evidence possibly by answering questions on these topics in the theory test and/or demonstrating your skills on these topics during the practical tests.

For this reason, the following information and/or notes are provided for you to read and use to ensure that you can satisfy the assessment requirements for this unit.

Work health and safety (WHS)

When you undertake the in-class simulated workplace tasks and/or the practical test for this unit (UEEEC0069), you're expected to demonstrate your knowledge of, and demonstrate your ability to carry out, the following performance criteria related to WHS and their associated knowledge evidence and performance evidence:

1.1 - Work health and safety (WHS) processes and workplace procedures for a given work area are obtained and applied (UEECD0007 references: 1.2, 1.3 & 1.4)

1.2 - WHS risk control work preparation measures and workplace procedures are followed (UEECD0007 references: 1.2, 1.3 & 1.4)

2.1 - WHS risk control work measures and workplace procedures are followed (UEECD0007 references: 1.4, 1.6, 2.1 and 2.4)

2.2 - Need to test and measure live work is determined in accordance with WHS requirements and workplace procedures (UEECD0007 references: 1.2, 1.3 & 1.4)

3.1 - WHS work completion risk control measures and workplace procedures are followed (UEECD0007 references: 3.1 to 3.4)

3.2 - Worksite is cleaned and made safe in accordance with workplace procedures (UEECD0007 references: 2.6)

Source: https://training.gov.au

As you may already be aware:

- WHS is taught and assessed in a dedicated unit, UEECD0007 (formerly UEENEEE101A)
- CD0007 or E101A is a pre-requisite unit for undertaking UEEEC0069 (that is, you must have successfully completed UEECD0007 or E101A to be able to undertake UEEEC0069)
- The performance criteria listed above are underpinned by, and/or are identical in substance to, the performance criteria from CD0007 (and E101A) indicated in the brackets above

Importantly, the knowledge and skills needed to perform these performance criteria are identical to those needed to troubleshoot digital circuits. This means that you already have the ability to demonstrate your knowledge of, and demonstrate your ability to carry out, these performance criteria in the context of this unit.

However, if your unsure and would like to read more information about these performance criteria, please refer to your learner resources for CD0007/E101A that specifically address them.

Note: If you have not successfully completed UEECD0007 or UEENEEE101A, please bring this to the urgent attention of the teacher before undertaking any of the simulated workplace tasks.

Using sustainable energy practices

When you undertake the in-class simulated workplace tasks and/or the practical test for this unit (UEEEC0069), you're expected to demonstrate your knowledge of, and demonstrate your ability to carry out, the following performance criterion:

2.6 - Fault-finding activities are carried out efficiently without waste of materials, or damage to apparatus, the surrounding environment or services using sustainable energy practices.

This includes knowing about sustainable energy principles and using sustainable energy practices.

Sustainability is a much used word now that describes growing global response to the short and long term negative effects on the planet of certain human activities, particularly in relation to climate change and pollution. It's accepted that, without drastic action, both problems will likely make living on this planet more difficult (due to extreme weather and toxic environments) that, in turn, risk causing drought, famine, war and the failure of democratic governments and even whole nation states.

R J Fuller (Deakin University, 2005) describes four principles that underpin sustainability:

- Futurity - The concern for future generations
- Environment - The concern to protect the integrity of eco-systems
- Equity - The concern for the poor and disadvantaged
- Participation - The notion that individuals can participate in decisions affecting them

An essential element of the global response to climate change is the switch to sustainable energy. Energy is said to be sustainable if it "meets the needs of the present without compromising the ability of future generations to meet their own needs." Underpinning this are the sustainable energy principles which include (but are not limited to):

- Reducing our use of non-renewable energy sources (like the fossils fuels which includes coal, gas and oil)
- Increasing our use of renewable sustainable energy sources (like solar just to name one of many).

Fuller's fourth principle is important because it means that individuals, including you, can contribute directly and indirectly to sustainable energy principles by using sustainable energy practices where possible. Direct action that can be taken includes (but is not limited to):

- Switching off equipment not in use
- Turning off lights when leaving a room with nobody in it
- Opening windows to cool rooms instead of turning on room air-conditioning
- Using natural lighting
- Minimising waste
- Recycling waste
- Using public transport to and from work
- Drinking tap water

Indirect action that can be taken, includes (but is not limited to) encouraging your employer to:

- Using more energy efficient lighting
- Using more energy efficient equipment
- Reducing (in winter) and increasing (in summer) the temperature setting of air conditioning to use less energy
- Switching to using sustainably sourced and/or sustainably manufactured materials
- Switching to an SEU energy provider

With all that said, every workplace is different so what works with one may not work with another.

Communication and documentation

When you undertake the in-class simulated workplace tasks and/or the practical test for this unit (UEEEC0069), you're expected to demonstrate your knowledge of, and demonstrate your ability to carry out, the following performance criteria related to workplace communication and documentation and their associated knowledge evidence and performance evidence:

1.3 - Scope of fault is obtained from documentation and/or work supervisor to determine the work to be undertaken

1.4 - Advice is sought from work supervisor to ensure work is coordinated effectively with others

1.5 - Sources of materials required for work are determined in accordance with workplace procedures

1.6 - Tools, equipment and testing devices required for work are obtained and checked for correct operation and safety.

3.3 - Work completion is documented and relevant person/s notified in accordance with workplace procedures

The specific knowledge required to satisfy these performance criteria is specific to each workplace setting. However, the skills are generic and so are transferable between workplace settings including simulated workplace settings at TAFE NSW. This means that, when you undertake the in-class simulated workplace tasks and the practical test, you are expected to demonstrate your skills in this regard.

Importantly, while undertaking the in-class simulated workplace tasks and/or the practical tests for this unit at TAFE NSW, the following applies to key terms in the performance criteria above:

- "Workplace supervisor" refers to the class teacher
- "Others" refers to the class teacher and other students in the class
- "Materials" refers to components and training aids needed to carry out the simulated workplace tasks as described in the skill practise exercises' equipment list
- "Workplace procedures" refer to the written and verbal procedures and instructions provided by the teacher for the safe and proper conduct of skill practise exercises
- "Tools, equipment and testing devices" refers to tools and test equipment needed to carry out the simulated workplace tasks as described in the skill practise exercises' equipment list and/or mounted on the workbenches
- "Documented... in accordance with workplace procedures" refers to the requirement for students to record their measurements, other results, and answers to questions on the skill practice exercises and/or any handouts that must be submitted to the teacher.

The following is the procedural requirement for undertaking the simulated workplace tasks that form the basis of skill practise exercises.

- Students must have permission from the teacher to start every skill practice exercise.
- Students must work individually on every skill practise exercise, or in pairs with the permission of the teacher (but never more than two per group).
- Students must perform a WHS risk assessment at the start of every skill practice exercise to ensure the safety of themselves and others.
- Students are to bring to the attention of the teacher any hazards that they identify for which there are they cannot implement appropriate control measures.
- Students are to collect equipment from the trolley in a safe and orderly manner having respect for the reasonable needs of other students.
- Students are to follow the written instructions in every skill practice exercises in correct order and in total including any qualifying notes.
- Students are expected to record their measurements, other results and answers to questions for every skill practice exercise and/or any handouts that must be submitted to the teacher.
- Students are expected to call the teacher to check their work when instructions direct them to do so.
- Students are expected to retake any measurements and reattempt and/or correct any of their answers to questions when reasonably directed to do so by the teacher.
- Students are not to return their equipment to the trolley or leave **until** they have completed the skill practice exercise and their work has been checked by the teacher.
- Students are expected to submit their work to the teacher by the due date when required to do so.
- At the completion of every skill practice exercise, students are expected to return all equipment to the trolley tidily (unless directed otherwise by the teacher) and ensure that the workspace is left cleaned and safe.
- Students are to bring to the attention of the teacher any faulty test equipment, tools and components.

Test and measurement devices and their safe use

When you undertake the in-class simulated workplace tasks and/or the practical test for this unit (UEEEC0069), you're expected to demonstrate your ability to safely test live circuits using voltmeters and logic probes.

You've never used logic probes before in your studies so you'll be taught how to do that.

However, you have been taught how to safely test live circuits and how to use DMMs as voltmeters, ammeters and ohmmeters to do so. Moreover, you have practised these skills multiple times in almost all units that you have undertaken so far. So, the following notes are provided in case you're unsure about how to do that and would like to read more information about these performance tasks.

Safety considerations when connecting test and measurement devices to circuits

When using a multimeter or oscilloscope to test and measure electronics equipment, there are two important safety measures that can minimise the risk of your being fatally electrocuted.

The first relates to the fact that technicians usually must necessarily test mains-operated electronics equipment (like a hi-fi or PVR) that is plugged-in, turned on and operating. To minimise the risk of electrocution, you must ensure that the equipment that you're testing or repairing is plugged into a power point that is protected at the switchboard by a residual current device (RCD). You can check whether the building's powerpoints are RCD protected by looking inside the switchboard. Figure 1 below shows an example of what to look for.

Figure 1

Up close, RCDs look like the example in Figure 2 below and there should be at least one row of these devices in the switchboard mounted side-by-side. RCDs are easily recognised by the fact that they're switched devices and often they have an in-built test feature too.

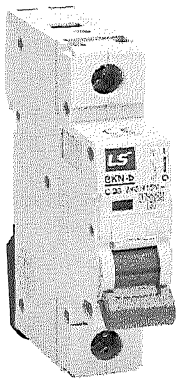


Figure 2

RCDs have been designed expressly for the purpose of preventing people and animals from a sustained direct connection to mains which can be fatal.

If the switchboard is older it may use semi-enclosed cartridge fuses instead of RCDs and an example is shown in Figure 3 below. Importantly, being fuses, these devices are designed to protect equipment and prevent fires but they **do not** prevent people/animals from potentially fatal electrocution. You should avoid powering the equipment that you're testing/reparing from power points that are only protected at the switchboard by these fuses.

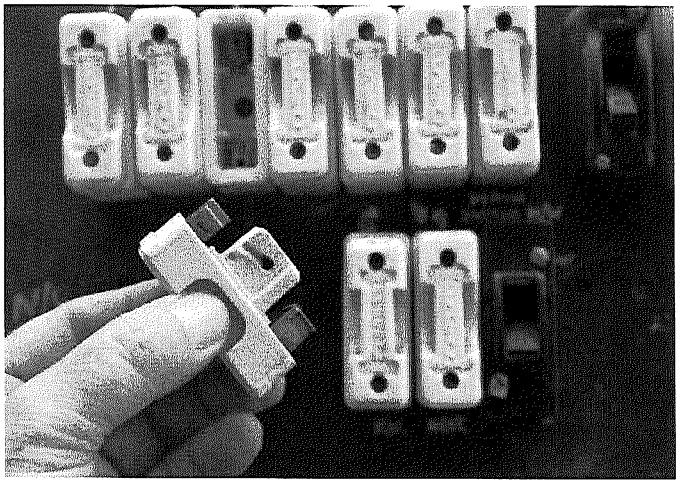


Figure 3

 Copyright: Not to be reproduced without permission

If you can't access the switchboard to see what's inside, or you know for certain that the power point is not RCD protected, then plug the equipment that you're testing into the powerpoint of a powerboard that is RCD protected instead. An example is shown in Figure 4 below - Notice the RCD at the left end (which includes a button for the test feature).

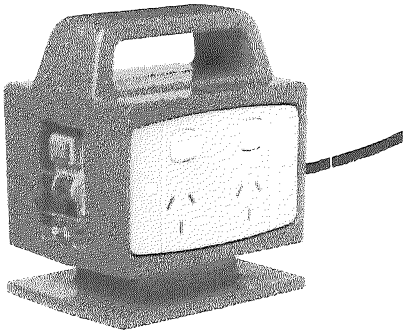


Figure 4

Take care when purchasing protected powerboards because there are many that offer "surge" and/or "overload" protection. While these features are useful, they don't provide protection from electrocution. Powerboards that are RCD protected will state this clearly and are usually a fair bit more expensive than simple surge and overload protected powerboards.

The second safety measure you must employ when connecting multimeters to live equipment, involves never doing so with your fingers touching any of the probe's metal parts. Figure 5 below shows the correct way to hold the probes. Notice that the person holds the probes behind the their slip barriers.

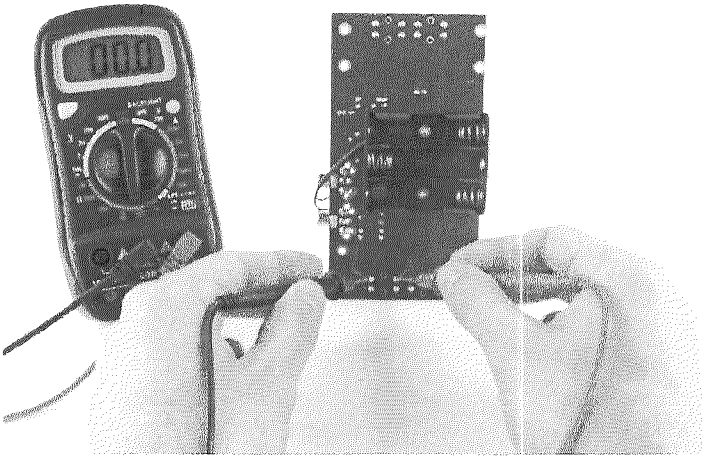


Figure 5

The correct connection of test and measurement devices to circuits

When measuring DC potential differences, the voltmeter is connected to the circuit in parallel with the component whose potential difference you want to measure. This is shown in Figure 6 below using a simple series circuit consisting of a resistor and a lamp with the meter measuring the DC potential difference across the resistor. The meter is shown both as a drawing (on the left) and as a schematic symbol (on the right).

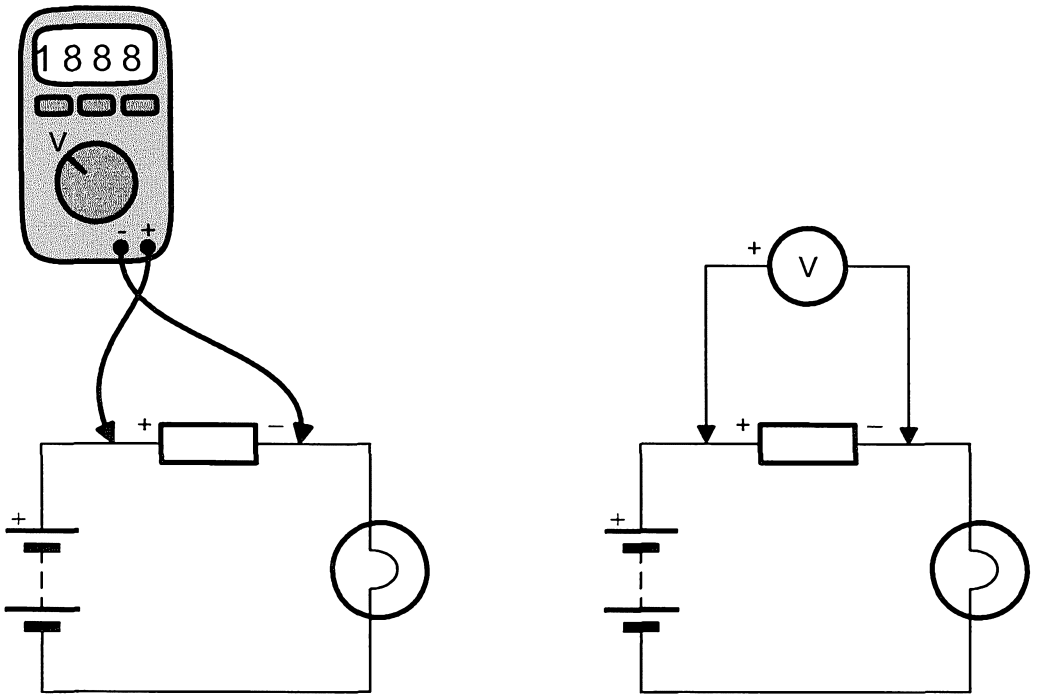


Figure 6

Notice that the voltmeter's positive probe must be connected to a DC potential difference's positive side. Importantly, although the electrical connection of the voltmeter is identical for measuring the potential difference in AC circuits (ie it's a parallel connection), you don't have to worry about which probe connects to which side of the component. This is because the direction of current is alternating (that is, it travels one way around the circuit for a time then reverses direction travelling the other way around the circuit for a time) and so the polarity of the potential difference (ie the voltage) is alternating too. For the instrument to cope with this, you must choose the multimeter's AC voltage measuring mode instead of its DC voltage measuring mode.

Care must be taken when connecting the meter to the circuit so that you don't actually short two points of the circuit together.

 Copyright: Not to be reproduced without permission

When measuring currents, the ammeter is connected to the circuit in series with the component whose current you want to measure. This is shown in Figure 7 below using a simple series circuit consisting of a resistor and a lamp with the meter measuring the current through the resistor (which is also the current through the lamp in this case because it's a series circuit). The meter is shown both as a drawing (on the left) and as a schematic symbol (on the right).

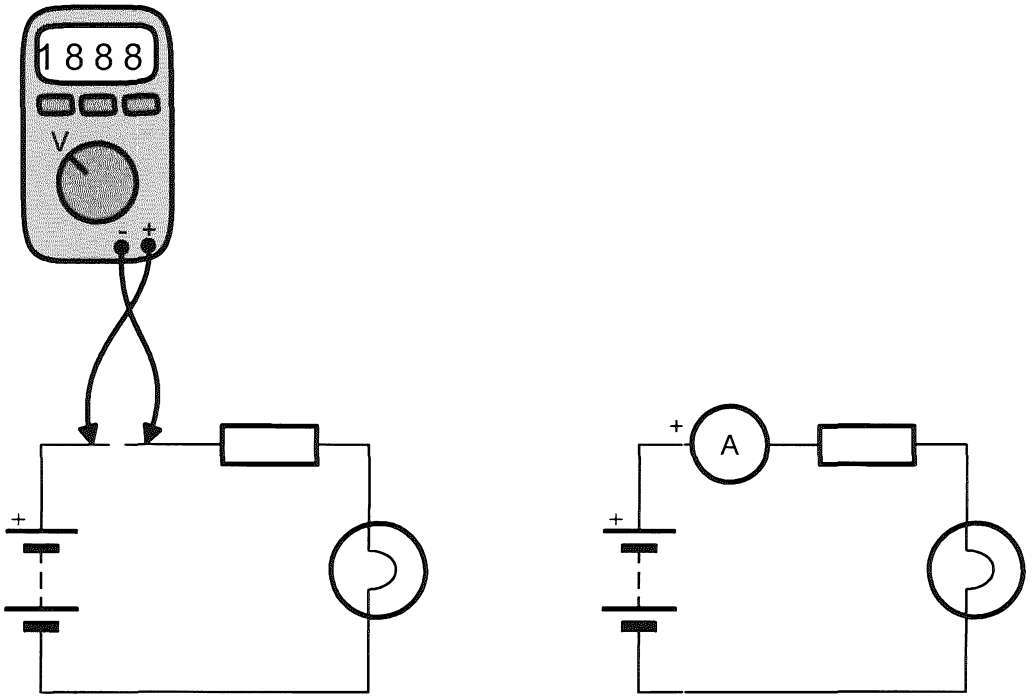


Figure 7

Notice that the circuit must be broken at some point to introduce the ammeter (an inconvenience that limits the practical measurement of current in the workplace). And, just like with AC voltage measurements, you don't have to worry about which probe to connect to which side of the break in the circuit. But you must choose the multimeter's AC current measuring mode instead of its DC current measuring mode.

Measuring resistance of resistors is a simple matter of connecting the meter's probes across the component in any polarity as shown in Figures 8 below. Importantly though, recall that the component should be removed from the circuit before you make the measurement.

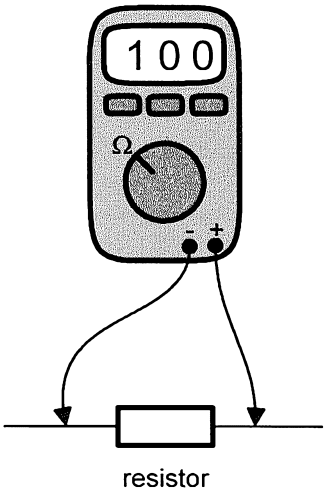


Figure 8a

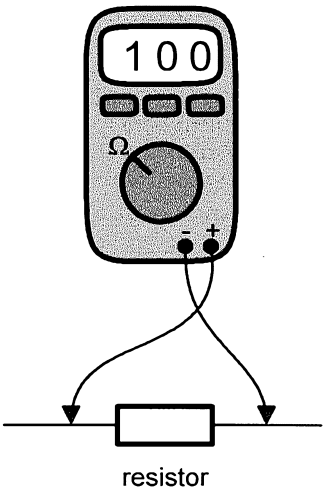


Figure 8b

Also remember that your fingers should not be touching the metal parts of the probes. If the component is out of the circuit, you're not going to get electrocuted. However, if you're touching the metal parts of both probes, the ohmmeter may show an incorrect resistance for the component because the measurement will include your bodily resistance.

Copyright: Not to be reproduced without permission

Student notes

Section 1 An introduction to digital electronics

Purpose To develop a basic understanding of digital electronics.

Objectives At the end of this section you should be able to:

- Explain the difference between physical properties that occur (or are represented) in *analog* and *digital* form
- Identify analog and digital signals
- Give examples of equipment in which you're likely to find digital signals
- State the advantages of digital electronics over analog electronics
- List the names used for the two logic states used in digital electronics
- State the voltages typically used for a logic-0 and a logic-1
- Count to 1111 in binary
- Name the IC packages typically used for digital integrated circuits
- Explain the two methods used by manufacturers of ICs to identify pin-1 on a DIL package digital IC
- Identify the pin number of all pins on a DIL package digital IC
- Explain *electrostatic charge*
- Name the methods by which electrostatic charges build
- Explain the main method by which electrostatic charges build
- Explain what happens during an electrostatic discharge (ESD)
- Explain why electrostatic discharge can damage electronics components
- List ways of minimising electrostatic discharge when repairing electronics equipment
- Use a DMM, logic probe and oscilloscope to test and measure a digital IC

Introduction

As you know, the world has gone digital! Digital mobile phones replaced analogue mobile phones well over a decade ago. DVDs have replaced video tape as the preferred medium in purchases of pre-recorded films by consumers without streaming. Since 20013 the free-to-air television stations here in Australia have only broadcast digital signals. Even radio stations here in Sydney are transmitting using the DAB+ digital protocol (though, there's no schedule for ending analog transmissions that your author is aware of).

While it seems that the world has suddenly discovered digital, the move to digital electronics has been happening steadily for over seventy years. As you probably already know, computers are digital machines but what you may not know is the first computers were developed in the late 1940's. Since then, personal computers (PCs) and there is now likely at least a PC or laptop in most Australian homes. But, it's not just in computers that you'll find digital electronics. Since the development of the microprocessor chip (in the early 1970's) digital electronics has become an integral part of most electronics equipment even equipment that is still analogue.

That being the case, as a service technician, it is essential for you to know a good deal about digital electronics.

What is digital electronics?

Before this question can be answered, the term *electrical intelligence* must be defined and the difference between analogue and digital properties must be described.

The term electrical intelligence is used in electronics as a name for electrical signals (that is, voltages and/or currents) that represent:

- information (such as numbers, letters or the condition of something)
- sounds (such as speech and music)
- pictures (such as drawings and photographs)
- video.

You'll find electrical intelligence inside most electronics equipment (for example, a radio or tv) assuming it's turned on and is working. An example of a voltage and/or current that is not electrical intelligence is *mains* (that is, the power out of a power point). Mains is used to power electrical and electronics equipment and does not represent information, sounds or pictures.

Turning to the difference between *digital* and *analogue* (hereafter spelt *analog* as this is common practice in the electrotechnology industry), the term digital has become an antonym for the analog way that most physical properties in the natural world behave (such as time, temperature and movement). To explicitly define the difference, we can say that:

> Analog properties can rest on an infinite number of values in a range whereas digital properties can rest on only a finite set of values in a range.

The effect of this difference is that analog properties can change in a continuous motion whereas digital properties can only change in steps. This might be better understood if we compare the difference between an analog and digital clock that both show hours and minutes.

As you know, the two hands on the analog clock rotate slowly around the clock face in a continuous movement. As such, if the clock's timing action is frozen at any time, it can be stopped at one of any of an infinite number of times in a 12 hour period (though our ability to visually distinguish between all those different times is fairly limited).

In contrast, the digital clock only changes between times in steps (for example, 1:00, 1:01, 1:02 and so on up to 12:59). So, if the digital clock's timing action is frozen at any time, it can be stopped at only one of 720 times in a 12 hour period.

Having defined electrical intelligence and described the difference between digital and analog properties, we can now return to the original question: What is digital electronics?

> Digital electronics is the class of electronic circuits/equipment that work with electrical intelligence in digital form.

What does digital electrical intelligence look like?

By definition, digital electrical intelligence or *digital signals* must have a finite number of discrete voltages or currents. A simple example of a digital signal is shown in Figure 1 below. The signal on the left is a sinewave (which, by nature, is analog) and the signal on the right is a digital equivalent of the sinewave. Notice that the digital signal's voltage doesn't change in a smooth (or continuous) movement like the sinewave. Instead, it changes in steps.

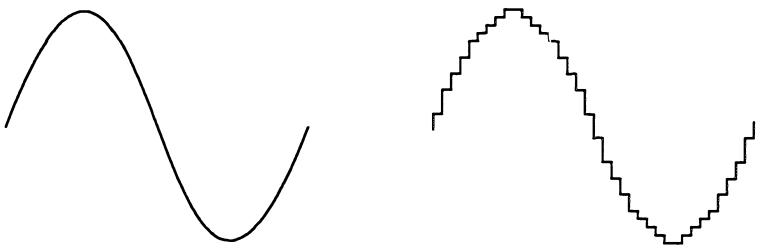


Figure 1 *Left) A sinewave which is an analog signal and; Right) A digital equivalent of a sinewave*

Despite the signal on the right in Figure 1 being a valid example of a digital signal, you wont see too many like this around digital electronics circuits (especially any equipment that is microprocessor controlled). That's because the signal consists of too many voltages.
As you will learn, digital electronics circuits (such as logic gates) are designed to work with pairs of voltages. That being the case, digital signals are more likely to look like a *pulse train* as shown in Figure 2 below.

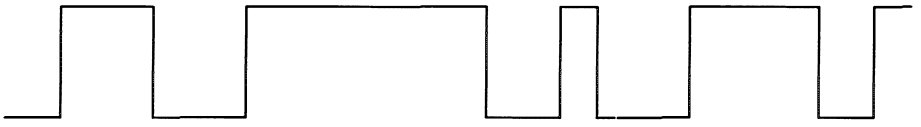


Figure 2 *The type of digital signal that you're likely to see around digital electronics circuits*

Logic states

Digital signals usually have just two conditions. These conditions are known as *logic states*. As Figure 3 below shows, one of the states is referred to variously as *logic-1*, *high*, *closed*, *set* and *on*. The other is referred to variously as *logic-0*, *low*, *false*, *open*, *reset* and *off*.

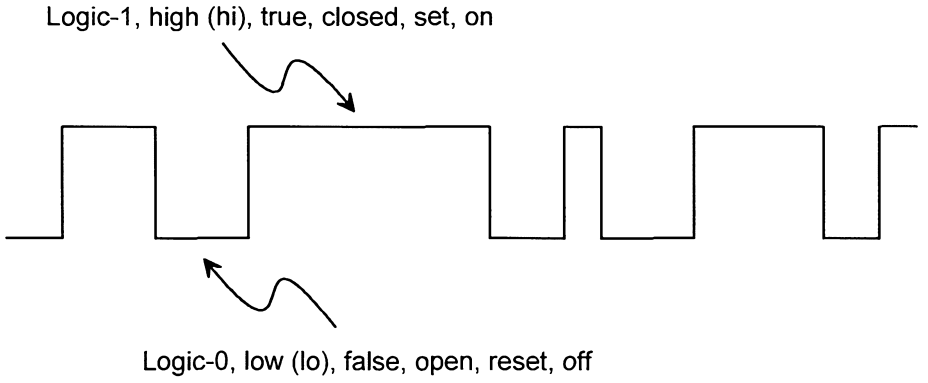


Figure 3 *Terms commonly used for the two conditions of digital signals*

Digital signal voltage levels

In a digital electronics circuit, the two logic states are physically represented by two different voltages. Two that are commonly used are: 0V for a logic-0 and 5V for a logic-1 (see Figure 4).

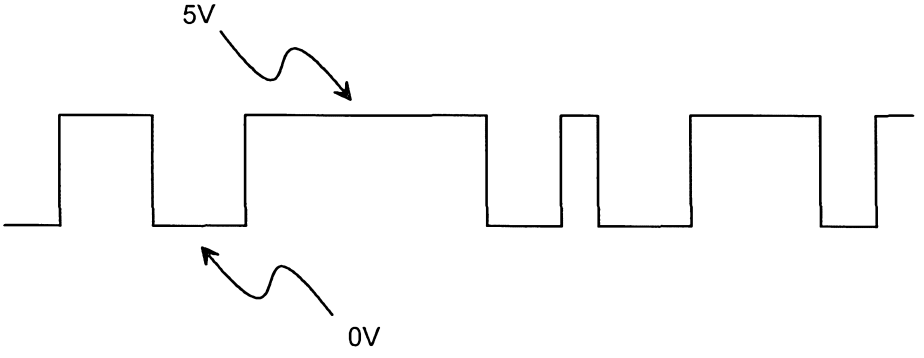


Figure 4 *0V and 5V are commonly used voltages for digital signals*

These voltages are ideal values but in practice they will vary a bit. Moreover, other pairs of voltages are used for digital signals so don't automatically assume a logic-0 and logic-1 to be 0V and 5V. This issue will be revisited later in this unit.

How can intelligence be represented by just two voltages?

At this point, it might seem odd that information, sounds and pictures can be represented by just two voltages, especially if you look back at the digital equivalent of a sinewave in Figure 1. It would need seventeen different voltage levels to make it up! But, remember that the two voltages physically represent logic-0 and logic-1 in a circuit. Moreover, logic-0 and logic-1 are numbers and, although they're small, they can be put together to make bigger numbers just like we do in decimal when making numbers bigger than nine (9). That being the case, a sinewave (or any other signal) can be converted into digital form by measuring its voltage at set intervals (a process called *sampling*) then converting the measured voltage into a number made of 0s and 1s. This is shown in Figure 5 below.

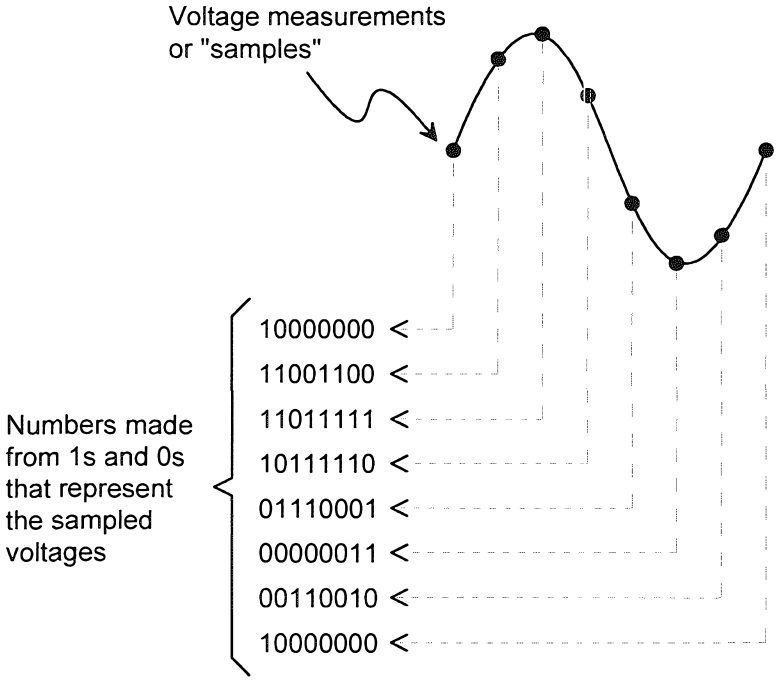


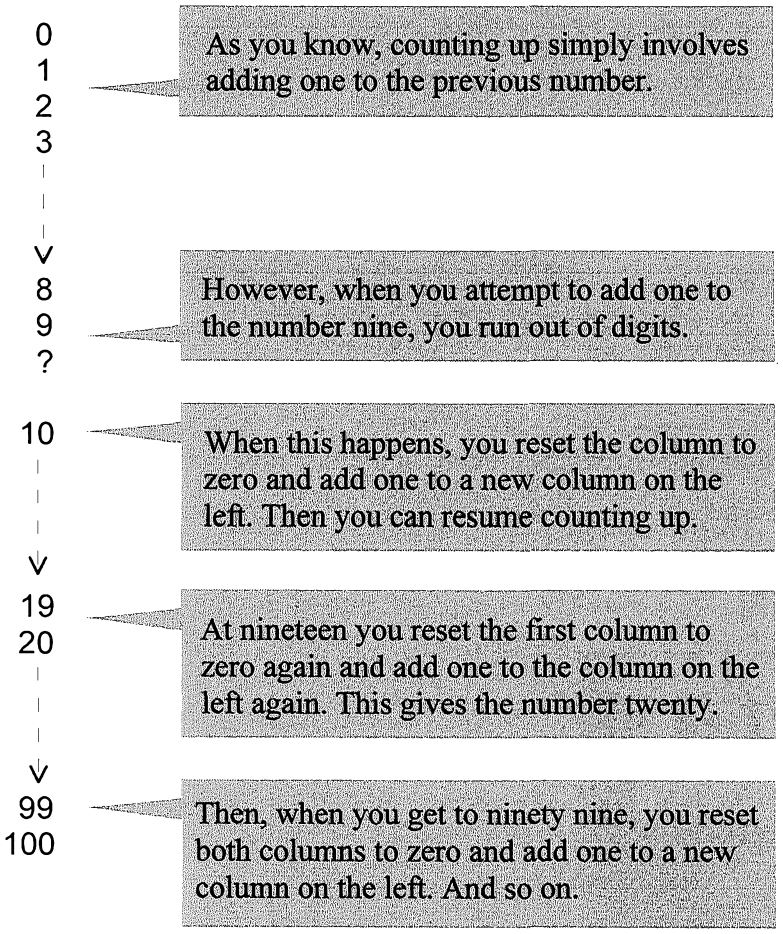
Figure 5 *A sinewave being converted to a digital form using only logic-0s and logic-1s*

The circuit that performs the conversion is called an *Analog-to-Digital Converter* (ADC) and these are used extensively to convert music into digital form for sale on compact discs (as an alternative to record albums which store the music in analog form).

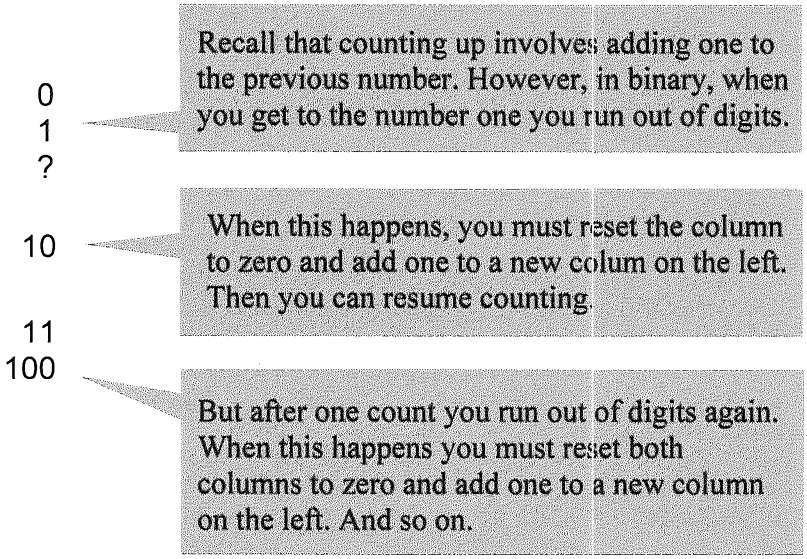
In addition, logic-0s and logic-1s can also be put together to form codes (instead of numbers) in much the same way as dots and dashes are used to represent letters and numbers in Morse code. There are several "binary" codes including *ASCII* and *BCD* and they lend themselves to applications such as converting alphanumeric information (that is, letters and numbers) into digital form.

Counting in binary

The number system that uses only the digits 0 and 1 is called *binary*. We'll leave learning more about the binary numbering system till later. However, at this stage you need to be able to count up in binary. This is not as hard as it might sound because the underlying principles for counting up in binary (or any number system for that matter) are exactly the same as for counting up in decimal. So, let's revise exactly what happens when you count up in decimal because you may not have thought about these principles for a long time.


0
1
2
3
8
9
?
10
19
20
99
100
As you know, counting up simply involves adding one to the previous number.
However, when you attempt to add one to the number nine, you run out of digits.
When this happens, you reset the column to zero and add one to a new column on the left. Then you can resume counting up.
At nineteen you reset the first column to zero again and add one to the column on the left again. This gives the number twenty.
Then, when you get to ninety nine, you reset both columns to zero and add one to a new column on the left. And so on.

Now let's apply the principles to counting in binary.


0
1
?
10
11
100
Recall that counting up involves adding one to the previous number. However, in binary, when you get to the number one you run out of digits.
When this happens, you must reset the column to zero and add one to a new colum on the left. Then you can resume counting.
But after one count you run out of digits again. When this happens you must reset both columns to zero and add one to a new column on the left. And so on.

The binary count sequence up to four columns is shown below:

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

Notice that all four columns have been used to write the numbers above even if four columns are not needed (for example, 0000 is used instead of just 0 and 0011 is used instead of just 11). These zeros are called *leading zeros* and it is necessary to show them in digital electronics because these are likely to be actual voltages on pins of an integrated circuit (IC) and circuit board tracks.

Advantages of digital electronics

There are many advantages of digital electronics over analog electronics. First, it is possible to store information electrically that was impossible or impractical to store previously. For good examples of this, consider text (writing) and photographs. Previous to digital electronics, this information could only be stored on paper (or, less efficiently, on clay, wood or rock).

Second, it is possible to store much more information in digital form than it is in analog (or other non-electrical forms). For an obvious example of this difference, compare the physical size of a standard record album (remember those big black discs?) which holds about 60 minutes of music and the much smaller compact disc which holds 74 minutes of music. For an even more dramatic example, consider the complete Encyclopedia Britannica. In paper form, it consists of 26 volumes and needs over a metre of bookshelf space. But, all of that information (including the photographs and diagrams) can fit on to a single CD ROM with room to spare.

Third, digital signals are much less susceptible to problems caused by electrical noise. This is one of the reasons why radio communications systems such as mobile phones use digital transmission systems.

And finally, thanks to microprocessors, information such as sound and pictures can be manipulated in digital form in ways impossible in analog (or at least not easily done anyway). For example, picture-in-picture allows you to watch one television channel (eg channel 7) and simultaneously view another channel (eg channel 9) as a little picture in the corner of the screen (this is called *picture-in-picture*). This was impossible for most of the time that analog television existed and, when it became possible, it was extremely expensive. Your author can't recall seeing this feature being available on analog televisions for sale in any electronics goods retailers.

What do digital circuit boards look like?

There can be big differences between the circuit boards of analog and digital circuits and an extreme example of each is shown in Figures 6 and 7 below. Figure 6 is the circuit board of a power supply and power supplies are almost totally analog devices. Figure 7 is a computer *motherboard* and, as we know, computers are almost totally digital machines.

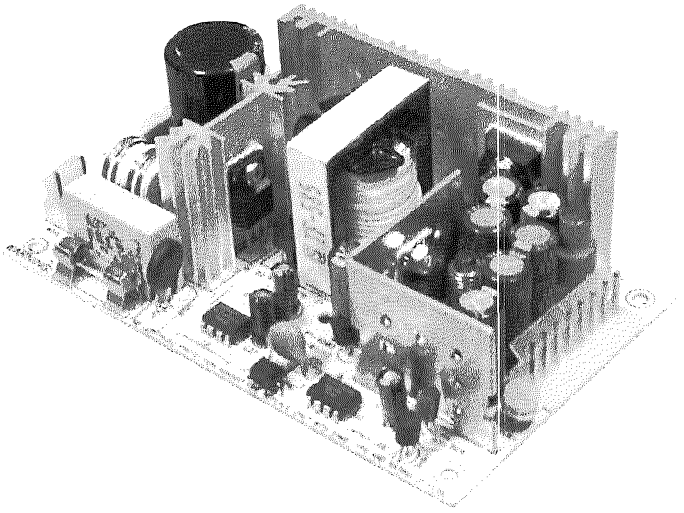


Figure 6 *The circuit board of a power supply*

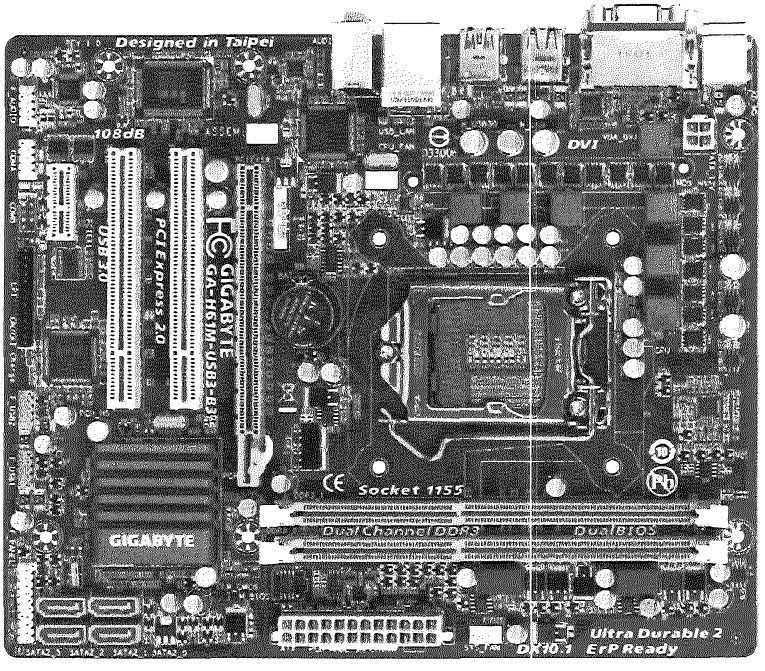


Figure 7 *A computer motherboard*

From these two examples we can see that, although analog circuit boards do have integrated circuits (ICs) mounted on them, they'll also often have a wide variety of other components including: resistors, capacitors, inductors, transistors, transformers and so on. Digital circuit boards on the other hand tend to have a lot less of these other components and many more ICs. Having said that, electronics equipment such as a hi-fi or a VCR are implemented using a mix of both analog and digital circuits so the circuit boards will have a more even share of ICs and other components.

Digital ICs

Figure 8 below is an example of a typical digital IC that you'll see inside digital electronics equipment. This kind of chip is called a *DIL* package. DIL is an acronym for ***dual in-line*** which is a reference to the pin arrangement. Another term you might see for this package is *DIP* which is an acronym for ***dual in-line package***.

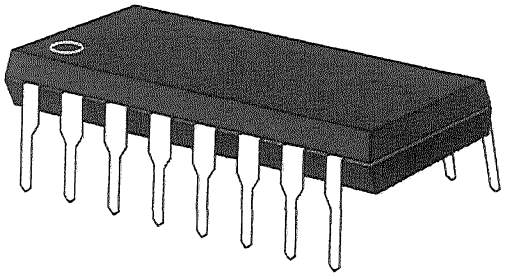


Figure 8 *A DIL or DIP package digital IC*

DIL package digital ICs typically have 14, 16, 18 or 20 pins though chips with 40 or more pins are used.

Figure 9 below shows the top view of a 16 pin DIL package digital IC. Notice that pin-1 is on the bottom left-hand side and the other pins are numbered from 2 to 16 in an anti-clockwise direction around the chip. This pin numbering system is used with all DIL IC packages. So, once you know which pin is pin-1, you automatically know the number of the other pins.

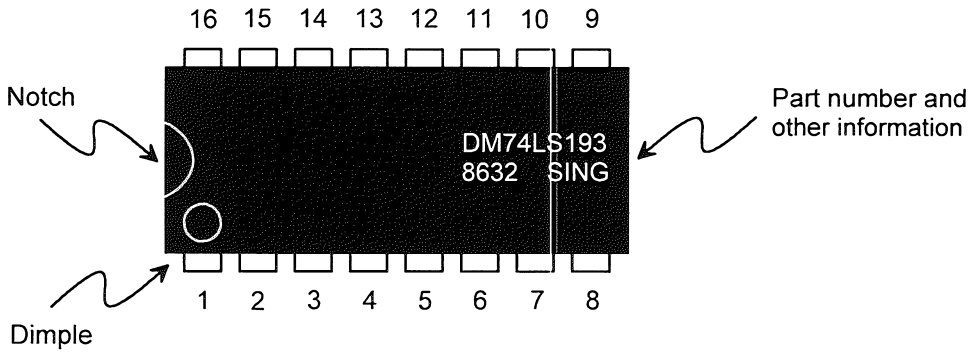


Figure 9 *Top view of a DIL package digital IC*

Pin-1 of DIL packages is usually indicated by one or both of two methods. One method is a *notch* (or U-shaped "cut-out") on one end. When the chip is oriented so that the notch is on the left hand side (as shown in Figure 9 above) the pin directly below it is pin-1. The other method is a *dimple* (or "dot" or "divot") directly above pin-1 (see Figure 9).

It is common for DIL package digital ICs to use the most right-hand pin of the bottom row (eg. pin-8 of a sixteen pin chip) and the most left-hand pin of the top row (eg. pin-16 of a sixteen pin chip) as the power supply connections (see Figure 10 below). However, always double-check the manufacturer's data before wiring the power supply to a chip that you've not used before because this isn't always the case.

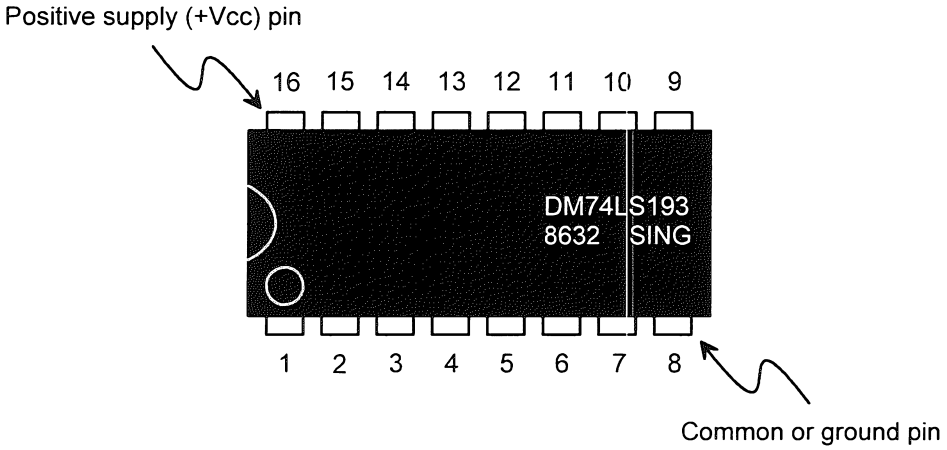


Figure 10 *The typical power supply connections to a sixteen pin DIL package digital IC*

Although the DIL package IC is typically between 20mm and 30mm long (depending on the number of pins) the chip inside the device is actually much smaller. Figure 11 below shows a DIL package IC that has been cut away to reveal the chip itself.

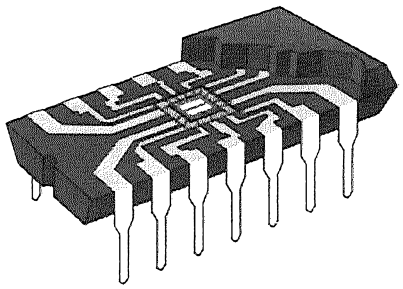


Figure 11 *A DIL package IC that has been cut away to reveal the chip inside*

The IC's package has been made much larger than the chip so that the people assembling the circuit boards can handle and solder the IC easily. However, in the past twenty years, the development of robotic printed circuit board assembly machines (called *pick-and-place* machines) has meant that it is no-longer necessary to make the packages so big. So, *surface-mount devices* (SMDs also called *SOIC* which is short for ***small outline integrated circuit***) were developed to take advantage of this to make circuit boards smaller. These ICs are less than half the size of DIL packages. Figure 12 below shows an example of a 16-pin SOIC package. With that said, these packages are still much larger than the actual integrated circuit inside.

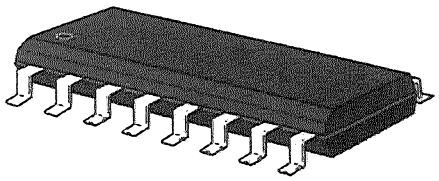


Figure 12 *A SOIC package*

A completely different package used for digital ICs with many pins is the PLCC which stands for *plastic leadless chip carrier*. Top and side views of this type of package are shown in Figure 13 below. Pin-1 is the pin with the notch above it and is often one of the pins in the middle of one side. The rest of the pins are counted off in an anti-clockwise direction.

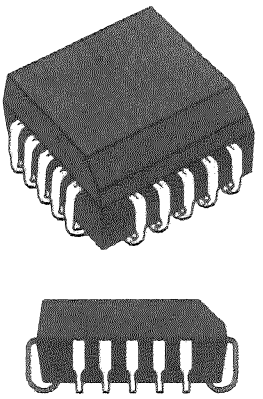


Figure 13 *Top and side views of the PLCC IC package*

Similar to the PLCC package is the LCCC package which stands for *leadless ceramic chip carrier*. An example is shown in Figure 14 below.

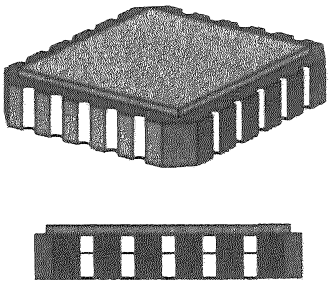


Figure 14 *Top and side views of the LCCC IC package*

Modern digital ICs often have a huge number of pins. For example, the latest Pentium processor has over four hundred! For these types of chips another IC package is used that is similar to the PLCC package except that all of the pins protrude from the bottom of the chip instead of the side. This type of package is called a PGA which stands for ***pin grid array***. Figures 15 and 16 show the top and bottom views of the PGA package.

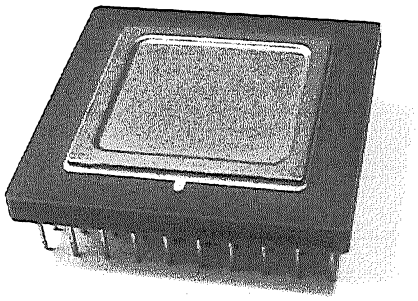


Figure 15 *Top view of a PGA package digital IC*

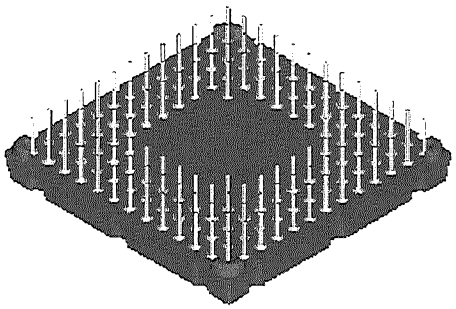


Figure 16 *Bottom view of a PGA package digital IC*

Electrostatic discharge

Electrostatic discharge (ESD) is the name for that irritating static "zap" that you get on a warm, dry day (often in spring!) when you touch something metallic like a car door or a door handle. However, ESD is more that just an irritation to the electronics industry because it can destroy or shorten the life of electronic components.

Consequently, manufacturers of electronic components and equipment take considerable effort to control ESD to prevent this kind of damage. But the issue is no less important in the repair workshop. ESD can further damage already faulty equipment by damaging additional components, increasing repair costs.

So, as a repair technician you will also need to take steps to control ESD and, to help you in this regard, you need to know a little bit about static electricity, how it's generated and how it discharges.

Charge and static electricity

Recall that a charge exists where there is an imbalance in the number of electrons between two points. When the imbalance of electrons occurs between the surfaces of any two materials (or on one surface of a non-conducting material) then the charge is called *electrostatic charge* or *static electricity* (see Figure 21 below).

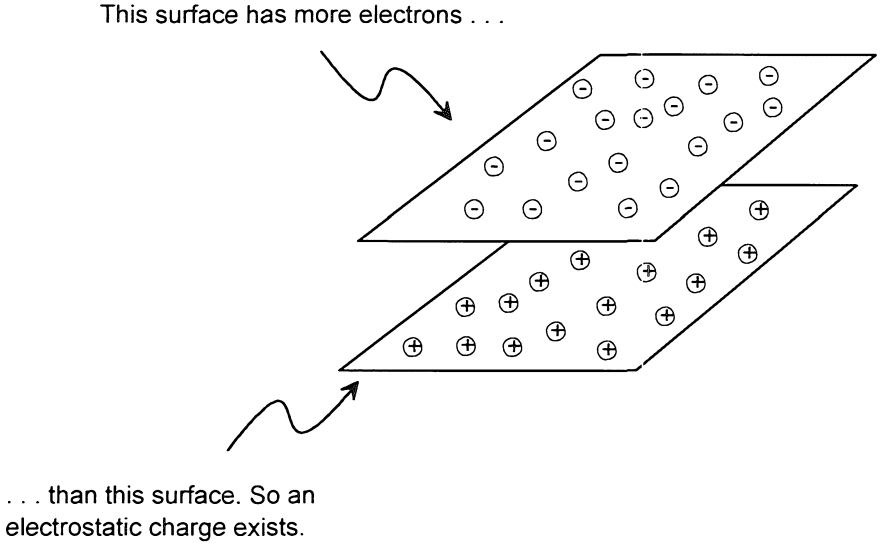


Figure 21 *An electrostatic charge exists where there is an imbalance of electrons between two surfaces*

As we know, if there is a conductive path between the two points where a charge exists then free electrons will move between them from the more negatively charged point to the less negatively charged point because opposite charges attract and free electrons can readily move between atoms. The same process occurs when a conductive path allows the imbalance of electrons in an electrostatic charge to equalise and this is what happens during an electrostatic discharge event.

Importantly, although you only tend to get ESD zaps on warm, dry days, in reality, you are continuously building up and discharging electrostatic charges. The reason you don't feel 99.999% of them is because they occur at currents below your threshold of feeling (which you might recall is 2mA). This is important because, even though you can't feel most of the electrostatic discharges that you're experiencing, the energy involved can potentially still be enough to damage electronics components.

How do electrostatic charges build up?

There are three basic methods that cause electrostatic charges to build up on people and objects. These are:

- triboelectric charging
- inductive charging
- capacitive charging

The most common form of these is triboelectric charging and a discussion of this follows. As the other two types of charging cause a much smaller number of ESD events and only occur under certain conditions, they are not discussed here.

The term *tribo* is from the Greek language and means "to rub" and this is exactly how triboelectric charges are built up. For example, as you take off your woollen sweater, the sweater rubs past your shirt and charges build up between the two (having said that, the action of taking off your sweater also allows these charges to be discharged and hence that crackle that you hear as you do).

During triboelectric charging, the objects being rubbed together come into contact with each other and separate many times. As they separate, electrons on one surface jump to the other. So, one surface can end up with more than its original share of electrons creating a charge between the two.

Virtually all objects can be a party to triboelectric charging (even air which explains why we get ESD zaps when simply walking outdoors). Whether an object looses or gains electrons depends on what the objects are made from. For example, wool would loose electrons (and hence become positively charged) when rubbed against polyester. But, wool would gain electrons (and hence become negatively charged) when rubbed against human hair. The number of electrons lost/gained is also determined by the types of materials. For example, wool would loose many more electrons when rubbed against polyester than it would when rubbed against wood.

As already mentioned, triboelectric charging is the most common cause of static electricity and can occur wherever objects can move. So even the action of ICs sliding back and forth inside a simple plastic IC tube during transportation can generate static charges between the ICs and between the ICs and the tube. In the same way, an electrostatic charge can build up between a circuit board and its packaging (which is why special anti-static packaging is usually used).

How does ESD cause electronics components to fail?

The voltages in ESD events can be quite high and can easily reach thousands of volts. But, as you have learnt, digital ICs tend to operate with voltages as low as 5V (and some as low as 3V). As the physical size of the components inside the chip are extremely small, it doesn't take a voltage much bigger than 5V (or 3V) on any of the chip's pins to cause metal contacts to melt, p-n junctions (which are the basis of diodes and transistors) to breakdown and the chemical composition of non-conducting materials to change (oxide failure). Obviously then, the voltages of ESD events can easily cause serious damage to chips and render them useless.

However, useless chips are usually readily identified because any equipment that they're installed in is unlikely to work properly (or at all). A potentially more serious problem occurs when ESD damages chips such that they work intermittently or reduces their operating life. Not only can this kind of problem add costs to the repair, it may also harm the workshop's reputation.

Controlling ESD in the workshop

Importantly, it's impossible to eliminate ESD in the workshop. However, it is possible to control it and minimise the potential for ESD to damage the ICs and circuits boards that you may handle. The following is a list of strategies that you can use to help in this regard. You may already be using some.

1. Wear a grounded wrist-strap while working on circuit boards.

   This is the main method used to minimise ESD events. The wrist-strap consists of a velcro cuff (that you wear around your wrist) and a cord that is used to connect the cuff to a ground circuit via a 1MΩ ¼W resistor.

   The wrist-strap is designed to keep the wearer's body at or near ground potential (0V). This helps to prevent the wearer from building up a charge that can discharge into a component and damage it. Having said that, wearing a wrist-strap doesn't prevent a charge that the component itself may hold from discharging into the wearer. It is for this reason that the 1MΩ resistor is used. It acts to limit the current involved in this kind of discharge and help to minimise the risk of damage.

2. Work on anti-static surfaces.

   When working on circuit boards and removing new components from their ESD protective packaging, place them on an anti-static surface such as an anti-static mat. These can be purchased for under $100. The mat looks much like an overly large mouse-pad and it's surface is conductive. Anti-static mats operate like wrist-straps in that they have a lead that connects to ground (again via a resistor) keeping the mat's surface at or near ground potential.

   Other options include anti-static shoes, work benches with in-built anti-static surfaces, anti-static floor mats and anti-static flooring materials. Given that these options are more expensive, the choice to use these types of ESD minimisation measures should be weighed against the cost of any potential ESD damage.

3. Use test equipment and tools that are grounded.

   Most test equipment are grounded so this is not a huge problem but it pays to check. Also, make sure that your soldering iron has a grounded tip. This can be checked by simply connecting an ohmmeter or continuity tester between the iron's tip and the earth pin on its three-pin plug. A low reading indicates the tip is grounded (but wait till the iron cools down before you try this!).

4. Look for the *ESD Association* awareness symbols.

   The symbol in Figure 21 on packaging indicates that the components, ICs or circuit boards inside are especially ESD sensitive. So, it's wise to use as many as the precautions listed above when removing and handling the contents.

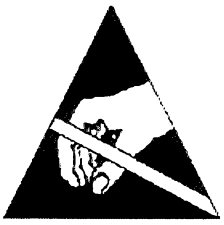


Figure 21

The symbol in Figure 22 indicates that the item is ESD protective. Equipment bearing this symbol should be used when handling electronics components.

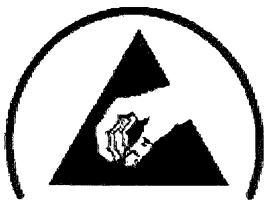


Figure 22

5. Treat all components as though they are especially static sensitive.

   Although it is true that some components are more sensitive to damage from ESD than others, all components are susceptible. As such, they should all be treated with care. Avoid touching any leads or pins on the device and don't handle components without a anti-static wrist strap if the day is warm and dry and you're experiencing lots of ESD zaps.

ESD myths

There are four common myths about ESD that need debunking.

1. You only discharge static electricity when you feel the static zap.

   Not true! Every time you move, you're building up and discharging static electricity. And surprisingly, you don't need to be dragging your feet along the carpet on a warm, dry day to generate destructive voltages. Simply shifting your position in your seat can generate hundreds of volts. The reason why you don't feel all of these ESD events is because they produce currents below our threshold of feeling. To be able feel an ESD event the current must be over 2mA and, at our body resistance (and wearing clothes), this requires voltages between 3,000 and 4,000 volts.

2. Only MOS devices (such as CMOS chips and MOSFETs) are susceptible to ESD.

   While it's true MOS devices are particularly susceptible to damage from ESD, all electronics components are susceptible, often at levels that we can't even feel. Table 1 below shows a list of components and the voltages at which they can be damaged.

Table 1

Components sensitive to voltages between 0 & 2kV	Components sensitive to voltages between 2kV & 4kV	Components sensitive to voltages greater than 4kV
▪ Microwave and very high frequency diodes & BJTs	▪ Protected MOS ICs	▪ Small signal BJTs (less than 5W)
▪ Many MOSFETs	▪ MOSFETs (eg CMOS)	▪ Small signal diodes
▪ Unprotected MOS ICs	▪ JFETs	▪ TTL logic
▪ Op amps with MOS capacitors	▪ Linear ICs (such as op amps)	▪ Quartz and piezoelectric crystals
▪ Many JFETs	▪ Hybrids	
▪ Precision diodes and regulators	▪ Low power BJTs (less than 100mW)	
▪ CCDs	▪ TTL with Schottky BJTs	
▪ Laser diodes	▪ ECL digital logic	
▪ Low current SCRs		
▪ Precision thin film resistors		

3. Only unmounted components are susceptible to ESD.

   Not true! The susceptibility of a component to being damaged by ESD doesn't change on or off a circuit board. Furthermore, unless the circuit board assembly has in-built ESD protection at critical junctions, components mounted on circuit boards are at greater risk of experiencing an ESD event. This is because the PCB tracks can carry an ESD event that occurs on one part of the board to components mounted all over it.

4. Only low humidity environments allow ESD events.

   While it's true that low humidity is the best environment for the build-up and discharge of static electricity, high humidity conditions can still produce ESD events involving relatively large voltages.

   Table 2 below shows a comparison of the types of voltages that can be produced for various activities at low and high relative humidity conditions. Notice that even in high humidity, the voltages involved are at levels that can damage or destroy many electronics components.

Table 2

Activity	Low humidity (10%-20% RH)	High humidity (70%-90% RH)
Walking across a vinyl floor	12kV	250V
Walking across synthetic carpet	35kV	1.5kV
Sitting on a foam cushion	18kV	1.5kV
Picking up a standard plastic bag	20kV	600V
Sliding a plastic box along a carpeted bench	18kV	1.5kV
Pulling tape off a PCB	12kV	1.5kV
Skin packing a PCB	16kV	3kV
Firing a solder sucker	8kV	1kV
Cleaning a PCB with an eraser	12kV	1kV
Freon circuit spray	15kV	5kV

Copyright: Not to be reproduced without permission

Student notes

Skill practice 1

An introduction to testing and measuring digital ICs

This exercise is a practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 1¼ hours to complete this exercise.

Equipment

- Digital trainer
- DMM
- logic probe
- 74LS193
- 470Ω ¼W resistor
- hook-up wire (terminated)
- IC extractor

Remember:

Follow TAFE NSW WHS guidance at all times!

 Copyright: Not to be reproduced without permission

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

Part A - Logic probe and digital trainer familiarisation

2. Gather the equipment needed for this exercise.

3. Turn on the digital trainer.

4. Turn on the digital multimeter and set it up as a voltmeter

5. Connect the voltmeter's black probe to the one of the trainer's ground sockets (they're marked *GND* and there's three of them, one on the right side and two on the left).

6. Connect the DMM's red probe to one of the trainer's eight data switches across the bottom of the trainer (marked *SW0* to *SW7*).

7. Set the switch you've chosen to the "0" position. Measure the switch terminal's output voltage. Record the voltage you measure in Table 1 below.

8. Set the switch you've chosen to the "1" position. Measure and record the switch terminal's new output voltage.

9. Connect the logic probe's power leads to the trainer's fixed 5V power supply.

   Note: To do this, use short terminated hook-up wires to connect the logic probe's red and black clips to the trainer's *+5V* and GND sockets (located directly under the trainer's on/off switch).

10. Set the two slide-switches on the logic probe so that they're in the *TTL* and *PULSE* positions.

    Note 1: TTL and CMOS are two widely used logic families that have different supply requirements and so the logic probe must be set for the family you're testing. The 74LS193 is a TTL IC.

    Note 2: You'll learn about the difference between the logic probe's *PULSE* and *MEMORY* modes later in this exercise.

11. Connect the logic probe to the output of the same switch as you chose for step 6.

12. Operate the switch and record the readings on the logic probe's LED display.

Table 1	Measured with DMM	Measured with Logic probe
Toggle switch in the "0" position		
Toggle switch in the "1" position		

 Copyright: Not to be reproduced without permission

Question 1
What is the advantage of using the logic probe instead of the DMM for measuring logic states?

13. Use hook-up wire to connect the same switch's output to one of the LEDs at the top right of the digital trainer via its socket which will be one of the set of eight sockets below the LEDs.

14. Operate the switch again.

Question 2
In what way are the LEDs like the logic probe and in what way are they not as good as the logic probe?

The teacher needs to check your work at this point...

15. Locate the push-button marked P_B at the bottom right hand corner of the digital trainer.

16. Locate this push-button's outputs which are to the left of the buttons and denoted using the letters B and $\overline{B}$.

17. Connect both of these outputs to an LED (that is, connect each output to their own LED).

18. Press, hold then release P_B and observe what happens to the LEDs.

19. Repeat step 18 a few times **without** holding the button down. Try pressing the button quickly **but** not so quickly that you risk damaging it.

Question 3
Briefly describe what you think you can use the two push-buttons for. **Tip:** Apart from "push", what other word do you think the "P" in P_B stands for?

Question 4
Based on your observations for step 19, what do you think would be a potential problem with using an ordinary LED as a logic probe for very short pulses?

20. Connect the logic probe to the P_B output.

21. Press and release the push-button while watching **all** of the logic probe's LEDs.

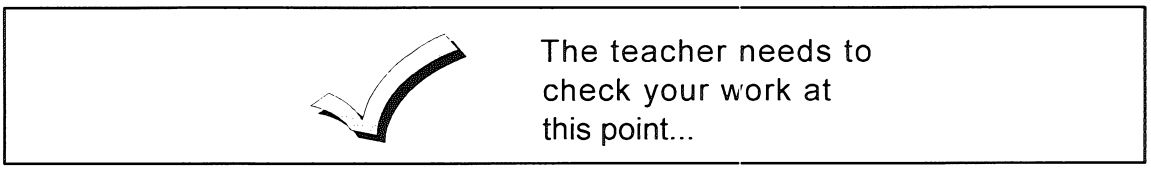
Question 5
What is the purpose of the yellow LED on the logic probe and when do you think it might be really useful based on your answer to Question 4?

 Copyright: Not to be reproduced without permission

22. While keeping the logic probe connected to the push-button's *B* output, set the appropriate switch on the logic probe to set it to *MEMORY* mode.

23. Press and release P_B again while watching the logic probe's LEDs.

Question 6
In what situations do you think would it be useful to use the logic probe's memory function?



24. Return the logic probe to *PULSE* mode.

25. Locate the trainer's in-built function generator (bottom left corner) and set its *Range* control to the *×1* position.

26. Turn the function generator's *Frequency* control to the "1" setting.

27. Connect the function generator's ***TTL Mode* output** to one of the trainer's LEDs.

Question 7
What type of AC waveform does the flashing LED suggest this signal is?

Question 8
As this signal is on the ***TTL Mode*** output, in what ways do you think it must be different to the squarewave on the trainer's function generator's other output (just marked "output")?

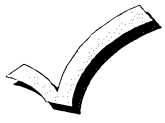
28. Slowly turn the *Frequency* control to the "10" setting and note what happens to the LED.

Question 9
What affect does this control have on the TTL signal?

29. Return the function generator's *Frequency* control to the "1" setting.

30. Set the function generator's *Range* control to the *×10* position and observe what happens.

Question 10
What affect does the *Range* control have on the TTL signal?



The teacher needs to check your work at this point...

 Copyright: Not to be reproduced without permission

Part B - Logic chip familiarisation

1. Turn off the digital trainer.

2. Wire the trainer's breadboard so that the +5V and ground outputs of its in-built power supply are connected to the breadboard as shown in Figure 1 below.

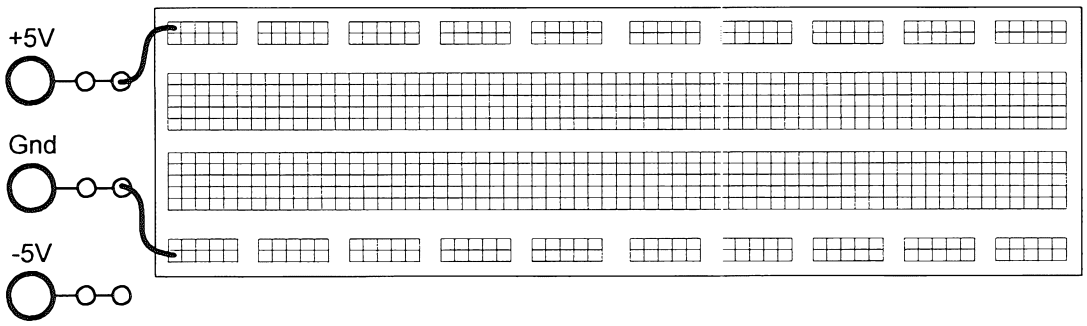


Figure 1

By wiring the breadboard in this way, you have set up +5V and ground "rails" along the length of the breadboard as shown in Figure 2. These can be used to connect power to multiple digital ICs.

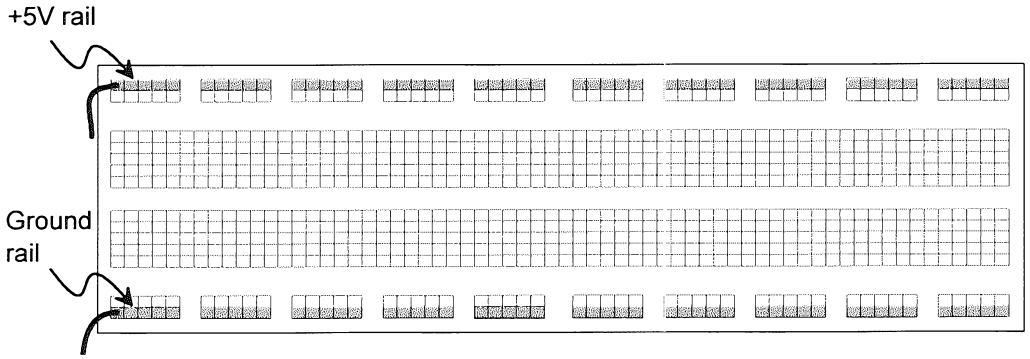


Figure 2

3. Insert the 74LS193 IC into the breadboard that you've just set up with power supply rails.

   Note: The chip must be inserted across the recess in the middle of the breadboard so that the pins on opposite sides of the chip are not connected to each other (see Figure 3).

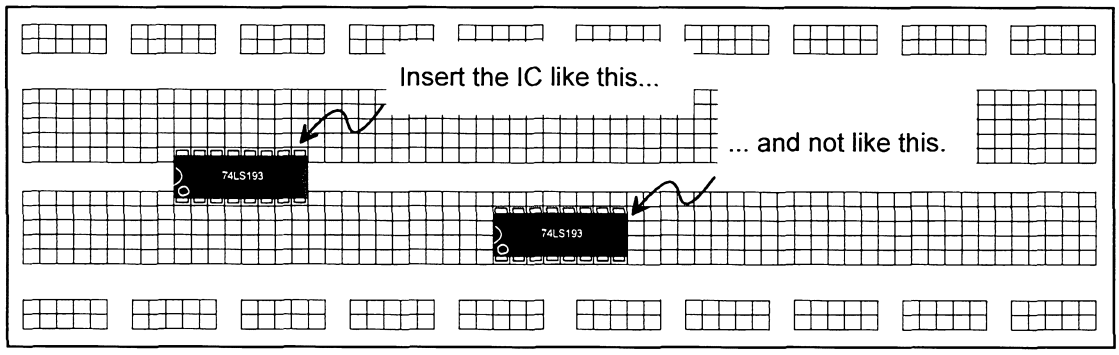


Figure 3

4. Connect pins **4, 11 and 16** of the chip directly to the +5V supply rail.

   Tip: When making connections to the pins of the IC, plug the hook-up wire into the furthest free hole away from the IC's pin. This is shown in Figure 4 below with the hook-up wire connecting pin-4 to the +5V supply rail. Connecting the hook-up wires in this way helps to keep the wiring neat and makes it much easier to remove the IC if necessary.

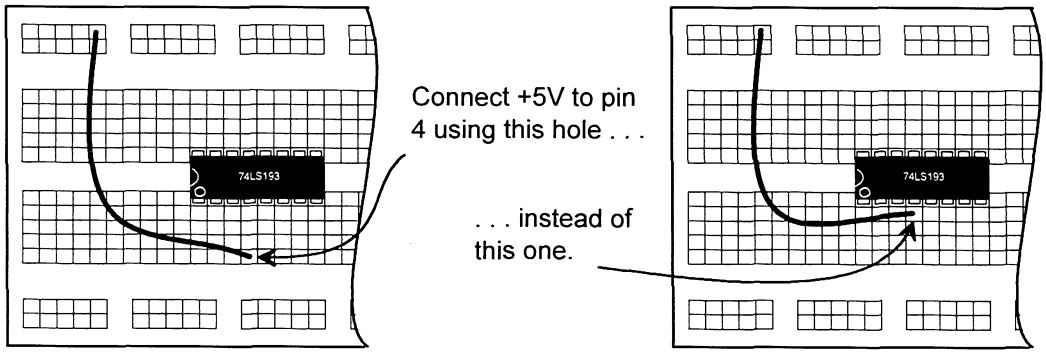


Figure 4

5. Connect pin-8 directly to the ground rail.

6. Connect pin-3 to LED 0.

7. Connect pin-2 to LED 1.

8. Connect pin-6 to LED 2.

9. Connect pin-7 to LED 3.

10. Connect a 470Ω resistor from pin-14 to the ground rail.

11. Connect pin-5 to the *B* output for P_B.

12. Turn on the digital trainer.

13. Insert one end of a medium length piece of hook-up wire into the +5V supply rail.

14. Use the other end of the wire to **momentarily** connect the IC's pin-14 to +5V.

    Note: The four LEDs on the trainer's display should now all be off indicating that the outputs are all logic-0. If they're not all logic-0 call the teacher.

15. Measure the voltages on pins 7, 6, 2 and 3 of the chip with respect to ground using the DMM. Record your measurements in Table 1 below.

Table 1

Voltages for logic-0			
Pin 7	**Pin 6**	**Pin 2**	**Pin 3**

Question 1
In digital electronics the ideal voltage for a logic-0 is 0V. Are your measured voltages close? Why do you think your measured values must be acceptable?

16. Record the logic levels on pins 7, 6, 2 and 3 of the chip in the first row of Table 2 below.

    Note: Recall that pin-7 is connected to LED 3, pin-6 is connected to LED 2, pin-2 is connected to LED 1 and pin-3 is connected to LED 0.

17. Press P_B **once only** and record the new logic levels on pins 7, 6, 2 and 3.

    Note: If none of the logic levels changed then call the teacher.

18. Repeat step 17 until Table 2 is full.

Table 2

Logic levels			
Pin 7	**Pin 6**	**Pin 2**	**Pin 3**

Question 2
From the results in Table 2, what is the chip doing every time you press P_B?

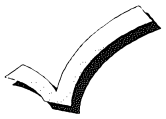
Copyright: Not to be reproduced without permission

19. With pins 7, 6, 2 and 3 all logic-1 measure the voltage on these pins with respect to ground using the DMM. Record your measurements in Table 3 below.

Table 3

Voltages for logic-1			
Pin 7	**Pin 6**	**Pin 2**	**Pin 3**

Question 3
In digital electronics the ideal voltage for a logic-1 is 5V. Are your measured voltages close? Why do you think your measured values must be acceptable?



The teacher needs to check your work at this point...

20. Set the *Range* control of the trainer's in-built function generator to the *×1* position.

21. Turn the trainer's *Frequency* control to the "1" setting.

22. Disconnect pin-5 of the chip from the output of P_B and connect the pin to the function generator's ***TTL Mode*** **output**.

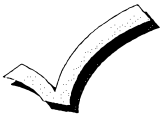
23. Describe what happens on the LED display.

24. Slowly turn the trainer's *Frequency* control to the "10" setting and observe the effect.

25. Set the trainer's *Range* control to the *×100* position and observe the effect.

Question 4
Why doesn't the LED display appear to be counting up any more?

Question 5
What type of waveform do you think you would see if you were to use a CRO to look at the signal on any one of the IC's output pins?



The teacher needs to check your work at this point...

Copyright: Not to be reproduced without permission

Student notes

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. Which of the three waveforms below is **NOT** a digital signal.

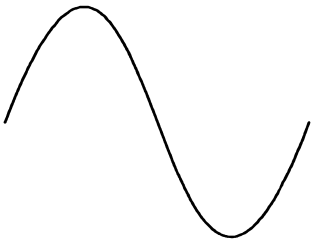


Figure 1

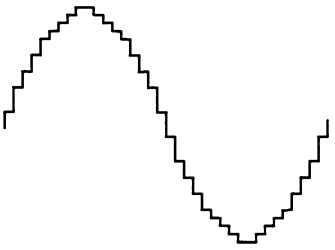


Figure 2

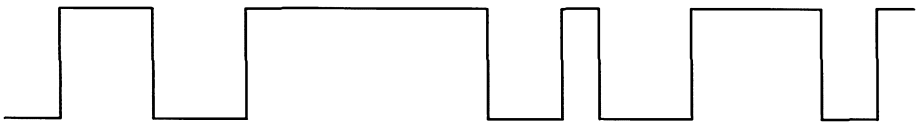


Figure 3

Copyright: Not to be reproduced without permission

2. Explain why the waveform in Figure 1 is an analog signal.

3. Explain why the waveforms Figures 2 and 3 are digital signals.

4. Explain why you're more likely to see the waveform in Figure 3 in digital electronics circuits.

5. List three types of equipment in which you're like to find digital signals.

6. Why is digital electronics being used instead of analog electronics for communications systems?

7. What are three other advantages of digital electronics over analog electronics?

8. On the digital signal in Figure 4, show which parts of the waveform are logic-0 and which parts are logic-1.

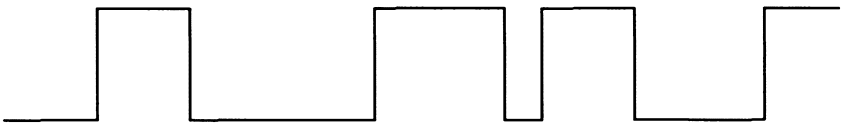


Figure 4

9. What are three other names for a logic-0?

10. What are three other names for a logic-1?

11. What are the ideal voltages typically used for a logic-0 and a logic-1?

12. What two IC packages are rectangular with pins on the long sides?

13. What two IC packages are square with pins on all four sides?

Copyright: Not to be reproduced without permission

14. What is the name of the IC package that is square with pins protruding from the under side?

_____

15. Identify pins 1, 9, 10 and 18 on the DIL package digital IC in Figure 5. **Note:** The IC has been drawn upside-down intentionally.

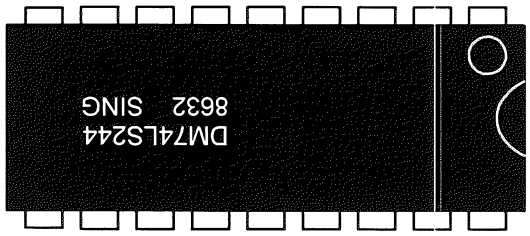


Figure 5

16. Why might a technician use a logic probe instead of a DMM when testing digital circuits?

_____

_____

17. What is the next number after:

a) 0100 _____

b) 1011 _____

c) 0111 _____

18. Electrostatic charge is an imbalance of

- ☐ atoms between two surfaces.
- ☐ protons between two surfaces.
- ☐ electrons between two surfaces.
- ☐ neutrons between two surfaces.

19. Most electrostatic charges are built up by a process of

- ☐ turboelectric charging.
- ☐ inductive charging.
- ☐ triboelectric charging.
- ☐ capacitive charging.

20. Briefly explain how triboelectric charging builds up electrostatic charges.

_____

_____

_____

21. Electrostatic discharge occurs when the imbalance of electrons on a surface

- ☐ is left alone.
- ☐ stays the same.
- ☐ gets worse.
- ☐ equalises.

22. Why can ESD damage digital ICs?

_____

_____

_____

23. Explain the two ways that an anti-static wrist-strap can prevent electronics components from becoming damaged by ESD.

_____

_____

_____

Copyright: Not to be reproduced without permission

24. What are four ways of minimising electrostatic discharge when repairing electronics equipment (other than wearing a wrist-strap)?

25. True or false: Every time you move, you build up and discharge electrostatic charges.

26. True or false: Most of the electrostatic charges that you build up and discharge are over 4kV.

27. True or false: Not many electronics components are damaged by ESD events up to 4kV.

28. True or false: Electronics components are more susceptible to damage from ESD on days with low humidity.

29. True or false: Electronics components are no more or less susceptible to damage from ESD when mounted on a circuit board without in-built static protection.

30. True or false: Electronics components are more likely to experience ESD when mounted on a circuit board without in-built static protection.

31. True or false: Wearing a wrist-strap will prevent ESD.

Section 2 Logic gates

Purpose To develop your ability to determine by measurement whether or not a logic gate is faulty.

Objectives At the end of this section you should be able to:

- List the seven logic gates
- Draw the distinctive-shape and rectangular-shape schematic symbols for the seven logic gates
- Identify logic gates in schematic diagrams
- Write the truth table for the seven logic gates
- Explain the operation of the seven logic gates without the use of a truth table
- Use pinout data to identify the input and output pins of logic gates inside digital ICs
- Use a DMM and logic probe to test and measure logic gate ICs and verify their operation

Introduction

Section 1 introduced you to the basic principles of digital electronics. You learnt that digital electronics is used in most electronics equipment these days and you learnt a little about digital signals, digital ICs and digital test equipment. This section introduces you to *logic gates* which are the building blocks of almost all digital electronics circuits.

Logic gates

Put simply, logic gates are circuits inside digital ICs that are designed to implement *logic functions*. A logic function is a type of mathematical operation like add or subtract. But, unlike regular mathematics, logic functions can only be performed on binary numbers (which is ideal for digital).

There are seven logic functions and they are:

- OR
- AND
- NOT
- NOR
- NAND
- Exclusive OR
- Exclusive NOR

For each logic functions there is a logic gate and, as a technician, you'll be expected to recognise logic gates in schematic diagrams and know how they operate so that you can test them when repairing digital electronics circuits.

The OR gate

The OR gate is a circuit that implements the OR logic function. It has one output and two or more inputs.

Schematic symbol

The *Institute of Electrical and Electronic Engineers* (IEEE) have a standard (IEEE Std 91-1984) that identifies three schematic symbols used for the OR gate and they are shown in Figure 1 below.

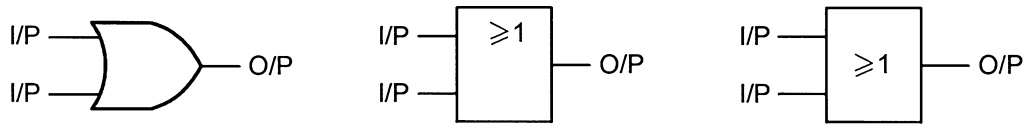


Figure 1 *The distinctive-shape and rectangular-shape schematic symbols for the OR gate*

It's a little inconvenient for students for there to be more than one schematic symbol for a device but, as all three are used, you are expected to know them all. To make things a little easier for you, the circuits in this module use the distinctive-shape symbols. However, you're expected to be able to recognise all three types of symbols for assessments.

Inputs to logic gates are often labelled alphabetically from A up. And outputs are usually labelled counter-alphabetically from Z down. So, the OR gate would usually be labelled as shown in Figure 2 below.

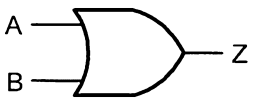


Figure 2 *The labelling of an OR gate's inputs and outputs*

Operation

The logic level on an input to any logic gate can only either be a logic-0 or a logic-1. That being the case, for a two input OR gate, there are four different possible combinations of zeros and ones that can be applied to them. These are 00, 01, 10 and 11. The OR gate's output is either a logic-0 or a logic-1 depending one which of these combinations is applied to the input.

The rule that determines the output of a two input OR gate is:

> The output is true (in other words a logic-1) when either input-A and/or input-B is true. When input-A and input-B are false, the output is false (logic-0).

This rule is demonstrated in Figure 3 below.

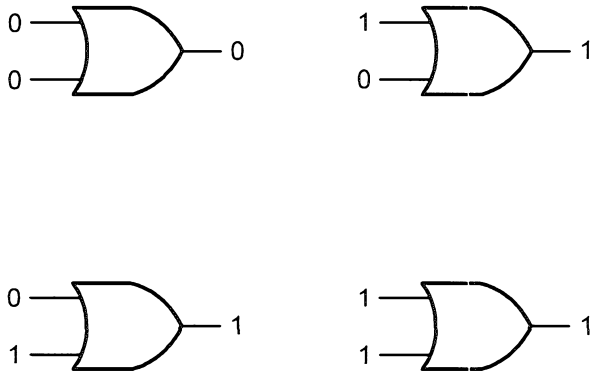


Figure 3 *The output of a two input OR gate for the four possible input conditions*

The operation of logic gates can be summarised in a table called a *truth table*. Table 1 below shows the truth table for an OR gate.

Table 1 *Truth table for an OR gate*

Inputs		Output
B	A	Z
0	0	0
0	1	1
1	0	1
1	1	1

A basic electrical equivalent of the OR function is shown in Figure 4 below. With both switches in the open position (remember that "open" is another term for logic-0), no current flows in the circuit and the lamp is off (and "off" is another term for logic-0). However, if either switch A or switch B is closed (remember that "closed" is another term for logic-1) current flows turning the lamp on (and "on" is another term for logic-1). And the lamp would be on if both switches are closed too.

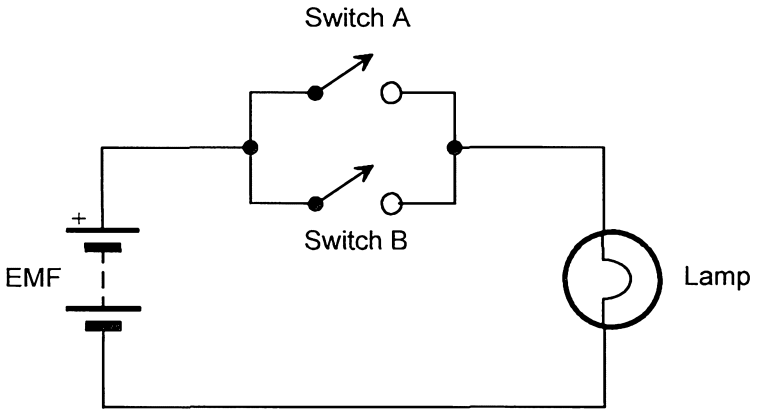


Figure 4 *A basic electrical equivalent of the OR function*

OR gate chips

Examples of digital ICs containing two input OR gates include the 7432 and the 4071. Digital ICs containing OR gates with more than two inputs are also available and include the 4075 (three input OR gates), the 4072 (four input OR gates) and the 4078 (an eight input OR gate).

Figure 5 shows an example of a three input OR gate and Table 2 shows its truth table.

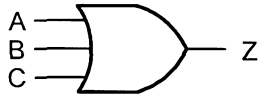


Figure 5

Table 2 *Truth table for a 3-input OR gate*

Inputs			Output
C	B	A	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Although the OR gate in Figure 5 has three inputs, its logic operation is the same as for the two input OR gate. That is, when any of the inputs are true, the output is true.

Application

An example of how an OR gate might be used in a digital electronics circuit is shown below in Figure 6. The circuit is a very simple alarm system consisting of two sensors (such as pressure mats or movement detectors) with output switches that are connected to the inputs of an OR gate. The OR gate's output is connected to the control input of an electronic switch. The switches in the sensors have a default position that has been hard-wired to a logic-0. When the sensor is activated the switch moves to the other position which has been hard-wired to a logic-1.

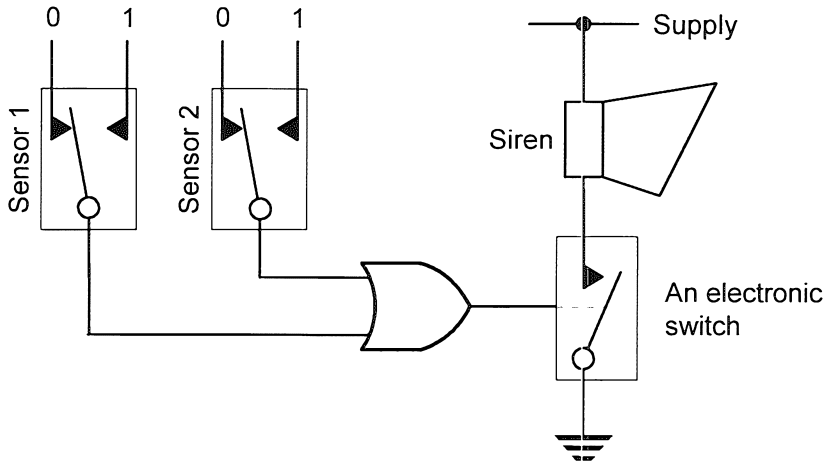


Figure 6 *The circuit of a very simple alarm system*

If there is no intruder, the sensors are not activated and both inputs to the OR gate are logic-0. As a result, the OR gate's output is also a logic-0 and so the electronic switch remains open and the siren is silent. These circuit conditions are shown in Figure 7 below.

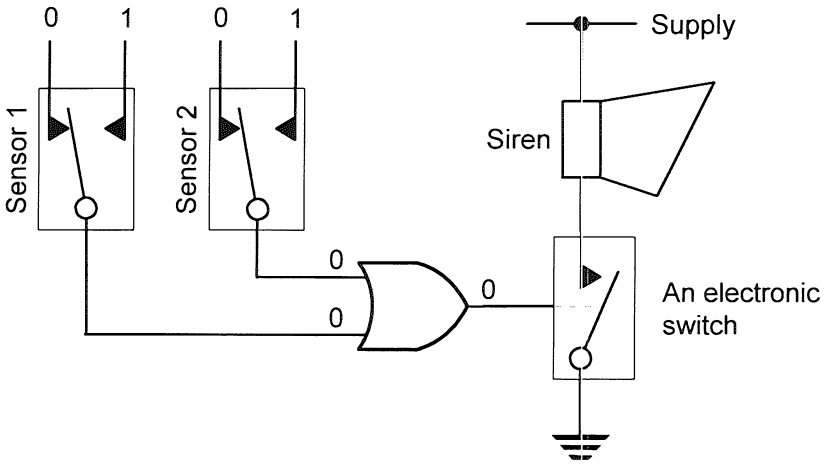


Figure 7 *The conditions around the circuit when there is no intruder*

However, if there is a break-in and sensor 1 detects the intruder its switch is activated forcing one of the OR gate's inputs to logic-1. This means that the OR gate's output will also go to logic-1 causing the electronic switch to close and the alarm to sound. These circuit conditions are shown in Figure 8 below.

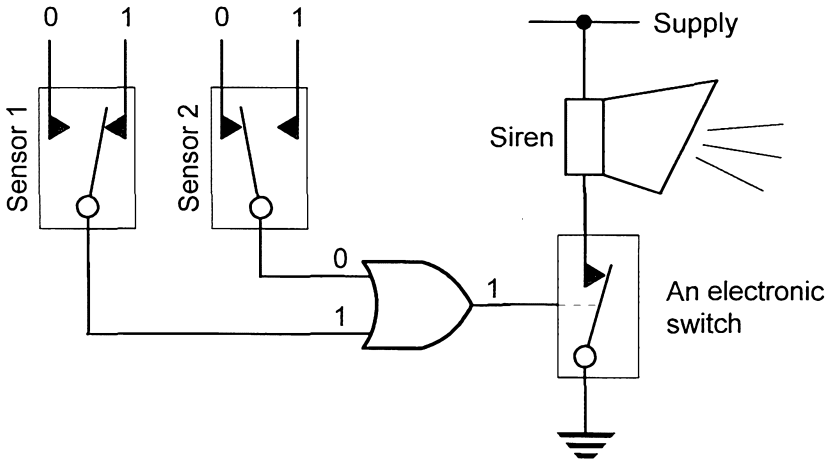


Figure 8 *The conditions around the circuit when an intruder is detected by sensor 1*

Importantly, if the intruder is detected by sensor 2 instead of sensor 1 the alarm will also sound (see Figure 9 below) and this is why the OR gate is used.

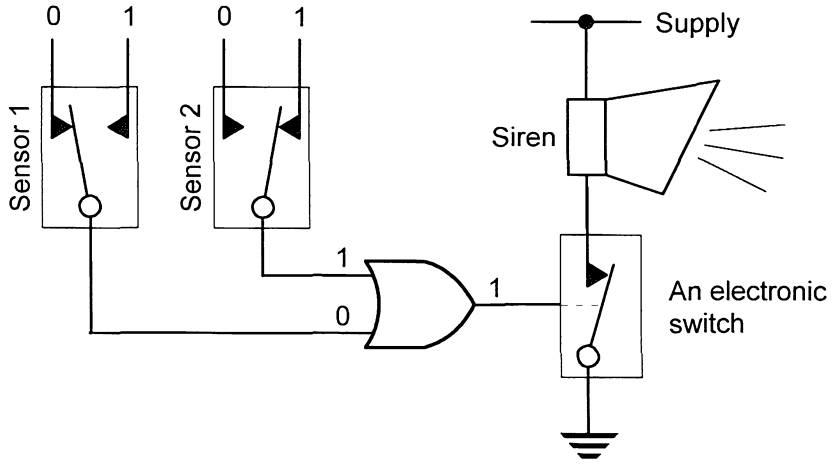


Figure 9 *The conditions around the circuit when an intruder is detected by sensor 2*

If the alarm system has more than two sensors, then an OR gate with the required number of inputs must be used (or a combinational logic circuit that implements an OR gate with the required number of inputs must be used - more later).

The AND gate

The AND gate is a circuit that implements the AND logic function. It has one output and two or more inputs.

Schematic symbol

Like the OR gate, there are three schematic symbols for the AND gate and they are shown in Figure 10 below. Remember, you must know all three of them as they are all used.

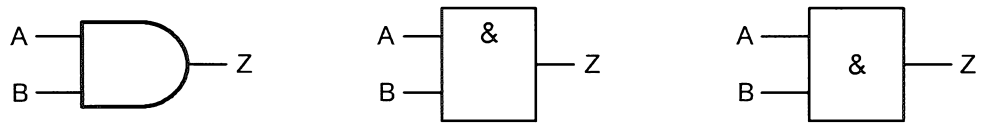


Figure 10 *The distinctive-shape and rectangular-shape schematic symbols for the AND gate*

Operation

The rule for the operation of a two input AND gate is:

> The output is true when both input-A **and** input-B are true. For all other input conditions, the output is false.

This rule is demonstrated in Figure 11 below.

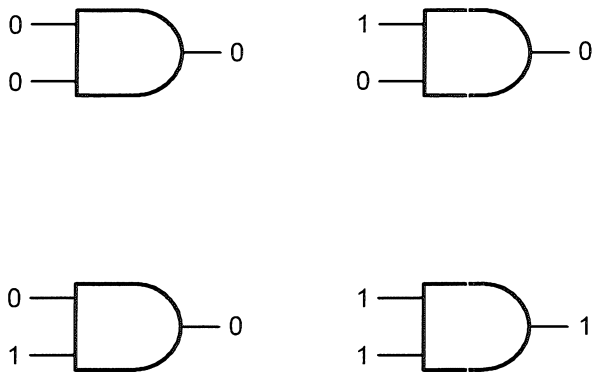


Figure 11 *The output of a two input AND gate for the four possible input conditions*

The AND gate's operation is summarised in the truth table in Table 3 below.

Table 3 *Truth table for an AND gate*

Inputs		Output
B	**A**	**Z**
0	0	0
0	1	0
1	0	0
1	1	1

A basic electrical equivalent of the AND function is shown in Figure 12 below. With either of the switches in the open position, no current flows in the circuit and the lamp is off. Only when both switch A and switch B are closed does the lamp turn on.

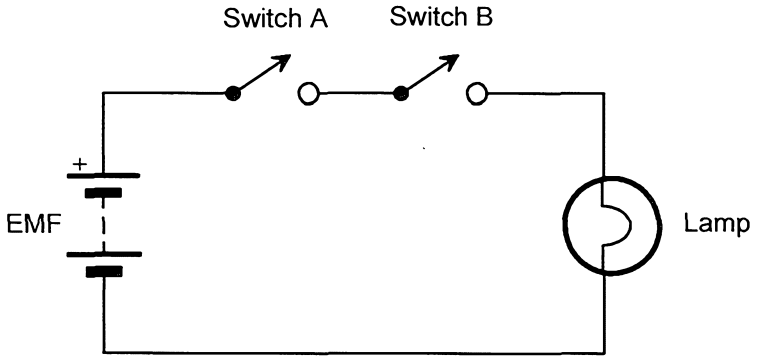


Figure 12 *A basic electrical equivalent of the AND function*

AND gate chips

Examples of digital ICs containing two input AND gates include the 7408 and the 4081. Digital ICs containing AND gates with more than two inputs are also available and include the 7411 (three input AND gates), the 7421 (four input AND gates) and the 4068 (an eight input AND gate).

Figure 13 shows an example of a three input AND gate and Table 4 is its truth table.

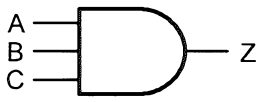


Figure 13

Table 4 *Truth table for a 3-input AND gate*

Inputs			Output
C	B	A	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Although the AND gate in Figure 13 has three inputs, its logic operation is the same as for the two input AND gate. That is, the output is true only when all three inputs are true.

Application

An example of how an AND gate might be used in a digital electronics circuit is shown below in Figure 14. This is another very simple alarm system consisting of one sensor that connects to one input of an AND gate. Connected to the gate's other input is a switch that toggles between either a logic-0 or a logic-1. This switch is used as a control that *arms* and *disarms* the alarm (that is, turns it on and off).

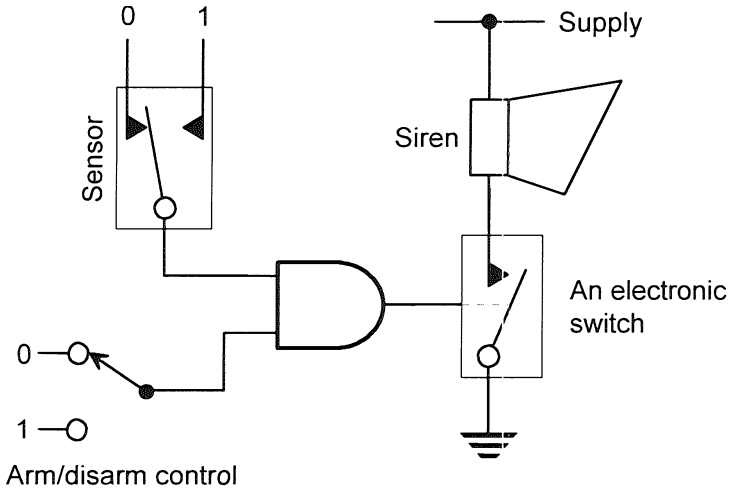


Figure 14 *A block diagram of a basic alarm system with an arm/disarm switch*

When the switch is in the position shown in Figure 15 below, a logic-0 is continuously applied to one of the AND gate's inputs. As we know from the AND gate's operation, this means that its output is a logic-0 regardless of the logic level on its other input. So, if the sensor detects a person in the room, the alarm isn't triggered. These circuit conditions are shown in Figure 15 below.

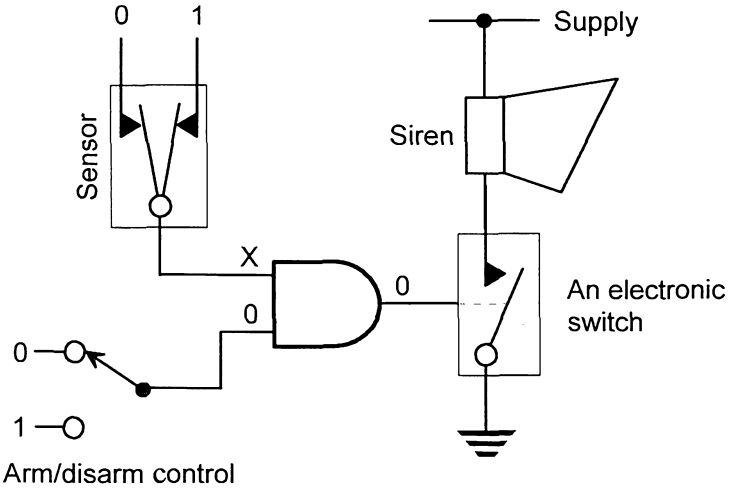


Figure 15 *When the alarm is disarmed the AND gate's output is logic-0 regardless of whether or not the sensor detects an intruder*

When the switch is in the position shown in Figure 16 below, a logic-1 is continuously applied to one of the AND gate's inputs. This means that the AND gate's output is the same as the logic level on its other input. So, if the sensor doesn't detect an intruder a logic-0 is applied to the AND gate and the siren doesn't sound. These circuit conditions are shown in Figure 16 below.

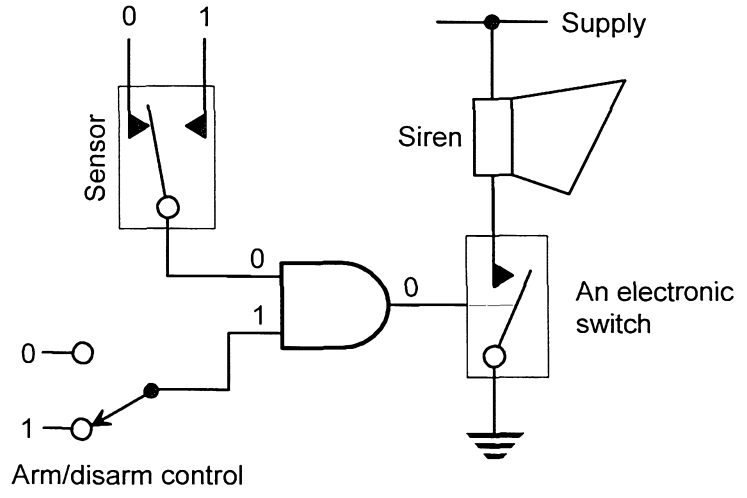


Figure 16 *The logic levels around the circuit when the alarm is armed but there is no intruder*

But, if the sensor detects an intruder, its output and the output of the AND gate changes to a logic-1 and the alarm sounds. These circuit conditions are shown in Figure 17 below.

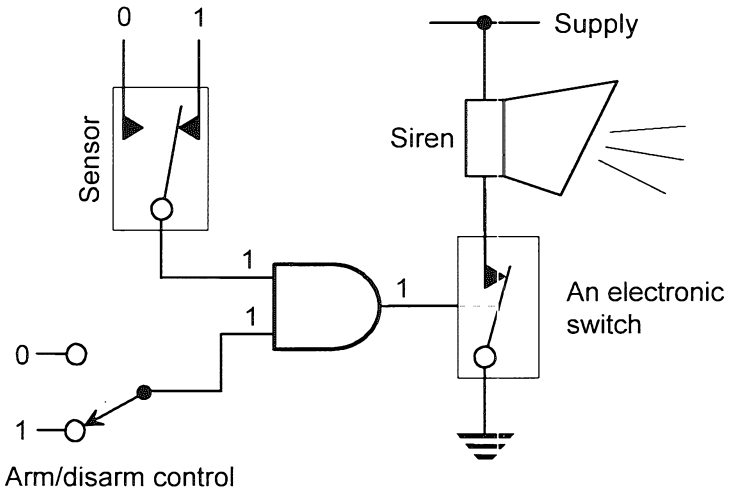


Figure 17 *The logic levels around the circuit when the alarm is armed and there is an intruder*

In real alarm circuits, the arm/disarm control would be a key operated or remote-control operated switch or a numerical keypad of some sort but the logic circuit is basically the same.

The NOT gate (also known as the Inverter)

The NOT gate is a circuit that implements the NOT logic function. It has one output and one input.

Schematic symbol

There are three schematic symbols for the NOT gate and they are shown in Figure 18 below. Remember, you must know all three of them as they are all used.

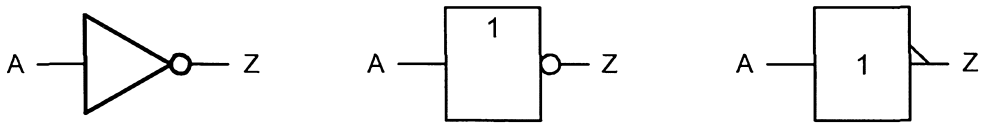


Figure 18 *The distinctive-shape and rectangular-shape schematic symbols for the NOT gate*

Operation

The rule for the operation of a NOT gate is:

The output is true when the input is false and the output is false when the input is true.

This rule is demonstrated in Figure 19 below.

Figure 19 *The output of a NOT gate for the two possible input conditions*

The NOT gate's operation is summarised in the truth table in Table 5 below.

Table 5 *Truth table for a NOT gate*

Input	Output
A	Z
0	1
1	0

A basic electrical equivalent of the NOT function is shown in Figure 20 below. With the switch in the open position, current flows in the circuit through the lamp and so it is on. But with the switch in the closed position, circuit current flows via the switch instead of the lamp and so the lamp is off. (Note, this model relies on the fact that a lamp has some resistance - say 40Ω - whereas the switch is a dead-short and so has no resistance and so all current would bypass the lamp.)

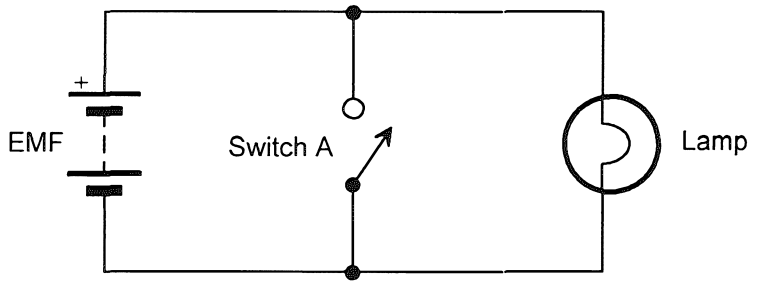


Figure 20 *A basic electrical equivalent of the NOT function*

NOT gate chips

Examples of digital ICs containing NOT gates include the 7404 and the 4009. It's not possible to have NOT gates with more than one logic input.

The NOR gate

The NOR gate is a circuit that implements the NOR logic function. It has one output and two or more inputs.

Schematic symbol

There are three schematic symbols for the NOR gate and they are shown in Figure 21 below. Remember, you must know all three of them as they are all used.

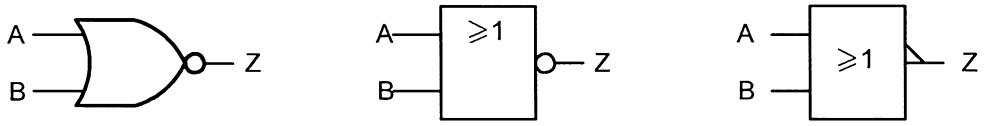


Figure 21 *The distinctive-shape and rectangular-shape schematic symbols for the NOR gate*

Operation

The rule for the operation of a two input NOR gate is:

> The output is false when either input-A or input-B is true. When input-A and input-B are false, the output is true.

This rule is demonstrated in Figure 22 below.

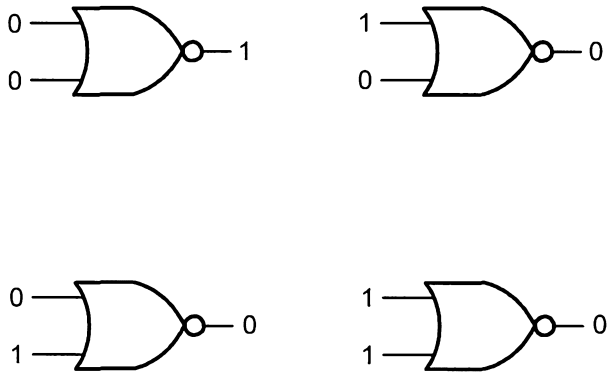


Figure 22 *The output of a two input NOR gate for the four possible input conditions*

The NOR gate's operation is summarised in the truth table in Table 6 below. If you compare the truth tables of the NOR and OR gates you'll notice that the logic levels on their outputs are the opposite of each other for the same input conditions.

Table 6 *Truth table for a NOR gate*

Inputs		Output
B	A	Z
0	0	1
0	1	0
1	0	0
1	1	0

A basic electrical equivalent of the NOR function is shown in Figure 23 below. When both of the switches are in the open position, current flows in the circuit through the lamp and so it is on. But if either of the switches are in the closed position, current flows through the closed switch instead of the lamp and so the lamp is off.

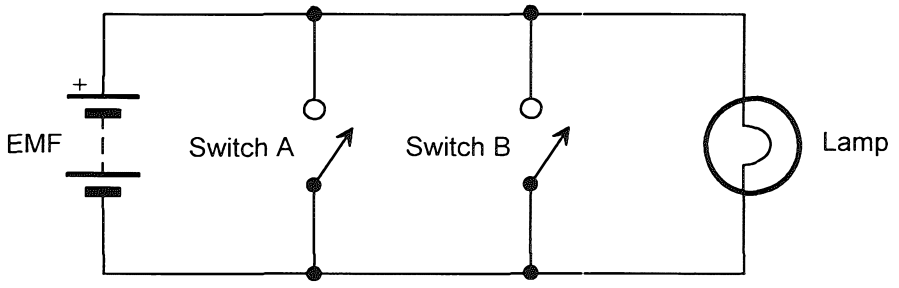


Figure 23 *A basic electrical equivalent of the NOR function*

NOR gate chips

Examples of digital ICs containing two input NOR gates include the 7402 and the 4001. Digital ICs containing NOR gates with more than two inputs are also available and include the 7427 (three input NOR gates), the 7425 (four input NOR gates) and the 4078 (an eight input NOR gate).

Figure 24 shows an example of a three input NOR gate and Table 7 is its truth table.

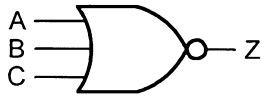


Figure 24

Table 7 *Truth table for a 3-input NOR gate*

Inputs			Output
C	B	A	Z
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Although the NOR gate in Figure 24 has three inputs, its logic operation is the same as for the two input NOR gate. That is, when any of the inputs are true, the output is false.

The NAND gate

The NAND gate is a circuit that implements the NAND logic function. It has one output and two or more inputs.

Schematic symbol

There are three schematic symbols for the NAND gate and they are shown in Figure 25 below. Remember, you must know all three of them as they are all used.

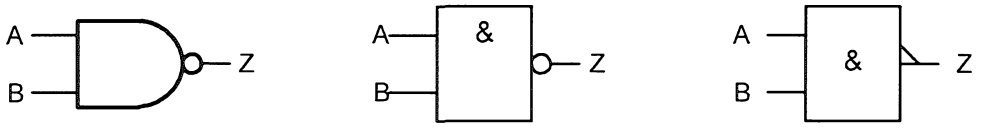


Figure 25 *The distinctive-shape and rectangular-shape schematic symbols for the NAND gate*

Operation

The rule for the operation of a two input NAND gate is:

> The output is false when both input-A and input-B are true. For all other input conditions, the output is true.

This rule is demonstrated in Figure 26 below.

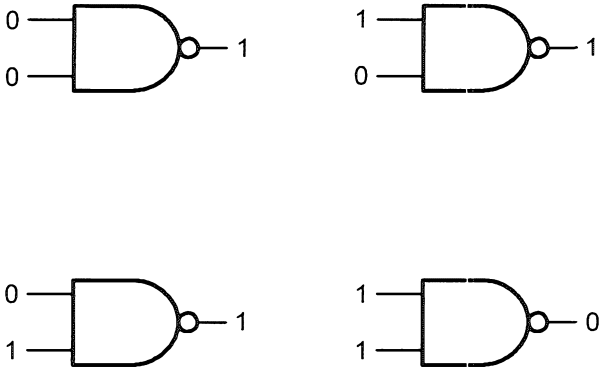


Figure 26 *The output of a two input NAND gate for the four possible input conditions*

The NAND gate's operation is summarised in the truth table in Table 8 below. If you compare the truth tables of the NAND and AND gates you'll notice that the logic levels on their outputs are the opposite of each other for the same input conditions.

Table 8 *Truth table for a NAND gate*

Inputs		Output
B	A	Z
0	0	1
0	1	1
1	0	1
1	1	0

A basic electrical equivalent of the NAND function is shown in Figure 27 below. When either of the switches are in the open position, current flows in the circuit through the lamp and so it is on. But if both of the switches are in the closed position, circuit current flows via the closed switches instead of the lamp and so the lamps is off.

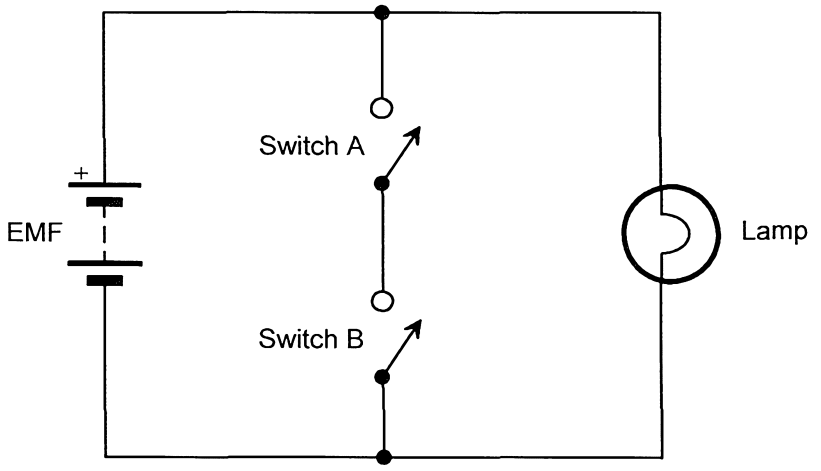


Figure 27 *A basic electrical equivalent of the NAND function*

NAND gate chips

Examples of digital ICs containing two input NAND gates include the 7400 and the 4011. Digital ICs containing NAND gates with more than two inputs are also available and include the 7410 (three input NAND gates), the 7420 (four input NAND gates), the 7430 (an eight input NAND gate) and the 74133 (a thirteen input NAND gate!).

Figure 28 shows an example of a three input NAND gate and Table 9 is its truth table.

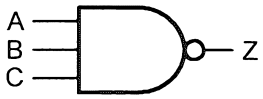


Figure 28

Table 9 *Truth table for a 3-input NAND gate*

Inputs			Output
C	B	A	Z
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Although the NAND gate in Figure 28 has three inputs, its logic operation is the same as for the two input NAND gate. That is, the output is false only when all three inputs are true.

The Exclusive OR gate

The Exclusive OR gate is a circuit that implements the Exclusive OR logic function. It has one output and two or more inputs.

Schematic symbol

There are three schematic symbols for the Exclusive OR (EXOR or just XOR) gate and they are shown in Figure 29 below. Remember, you must know all three of them as they are all used.

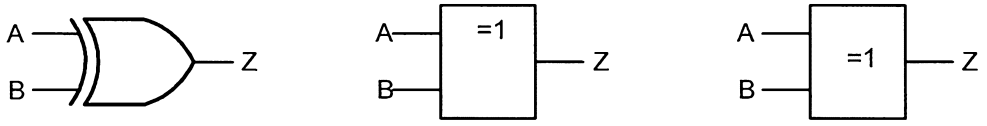


Figure 29 *The distinctive-shape and rectangular-shape schematic symbols for the XOR gate*

Operation

The rule for the operation of a two input XOR gate is:

> The output is true when inputs A are B are different. When inputs A and B are the same, the output is false.

This rule is demonstrated in Figure 30 below.

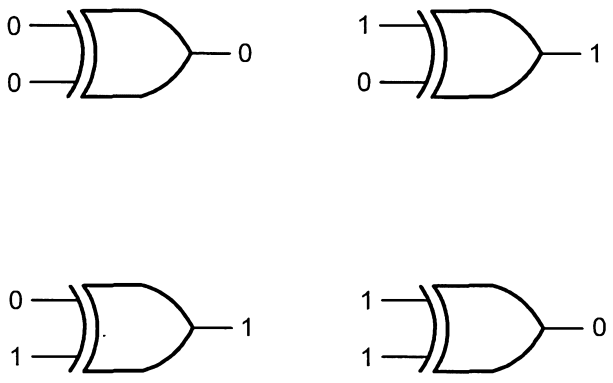


Figure 30 *The output of a two input XOR gate for the four possible input conditions*

The XOR gate's operation is summarised in the truth table in Table 10 below.

Table 10 *Truth table for an XOR gate*

Inputs		Output
B	A	Z
0	0	0
0	1	1
1	0	1
1	1	0

An electrical equivalent of the XOR function is not shown because it is more complicated than the electrical equivalents of the other logic functions we've been looking and would not necessarily help you to understand the XOR function. However, you have seen the circuit in an earlier subject because it's the one that allows two switches to control one light bulb (such as a light in a stairway with a switch at the top and bottom).

XOR gate chips

Examples of digital ICs containing two input XOR gates include the 7486 and the 4070. Although it would be possible to make an XOR gate with more than two inputs, digital ICs containing such devices are not available in the 74 and 4000 series of digital ICs.

The Exclusive NOR gate

The Exclusive NOR gate is a circuit that implements the Exclusive NOR logic function. It has one output and two or more inputs.

Schematic symbol

There are three schematic symbols for the Exclusive NOR (EXNOR or just XNOR) gate and they are shown in Figure 31 below. Remember, you must know all three of them as they are all used.

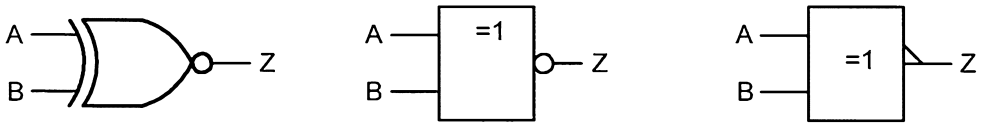


Figure 31 *The distinctive-shape and rectangular-shape schematic symbols for the XNOR gate*

Operation

The rule for the operation of a two input XNOR gate is:

> The output is true when inputs A and B are the same. When inputs A and B are different the output is false.

This rule is demonstrated in Figure 32 below.

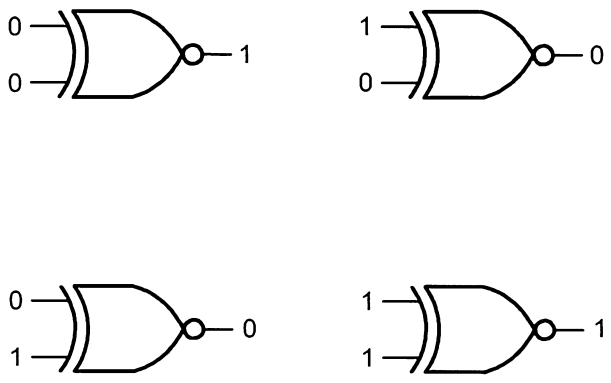


Figure 32 *The output of a two input XNOR gate for the four possible input conditions*

The XNOR gate's operation is summarised in the truth table in Table 11 below. If you compare the truth tables of the XOR and XNOR gates you'll notice that the logic levels on their outputs are the opposite of each other for the same input conditions.

Table 11 *Truth table for an XNOR gate*

Inputs		Output
B	A	Z
0	0	1
0	1	0
1	0	0
1	1	1

An electrical equivalent of the XNOR function is not shown because it is more complicated than the electrical equivalents of the other logic functions we've been looking and would not necessarily help you to understand the XNOR function.

XNOR gate chips

Examples of digital ICs containing two input XNOR gates include the 74266 and the 4077. Although it would be possible to make an XNOR gate with more than two inputs, digital ICs containing such devices are not available in the 74 and 4000 series of digital ICs.

Summary

Table 12 below is a summary of the operation of the logic gates (excluding the NOT gate).

Table 12

		OR	NOR	AND	NAND	XOR	XNOR
B	**A**	**Z**	**Z**	**Z**	**Z**	**Z**	**Z**
0	0	0	1	0	1	0	1
0	1	1	0	0	1	1	0
1	0	1	0	0	1	1	0
1	1	1	0	1	0	0	1

The following are some general rules that also describe the operation of logic gates.

1. For the AND gate - Any logic-0 on an input gives a logic-0 on the output.
2. For the OR gate - Any logic-1 on an input gives a logic-1 on the output.
3. For the NAND gate - Any logic-0 on an input gives a logic-1 of the output.
4. For the NOR gate - Any logic-1 on an input gives a logic-0 on the output.
5. The "bubble" is used to denote the invert or NOT function.

The pinout of common logic gate digital ICs

As already mentioned, digital ICs are available that contain one or more of each of the seven logic gates. When repairing digital electronics circuits you'll need to test the operation of gates inside digital ICs. To help you do this, manufacturers of digital ICs provide *pinout* data which tells you which pin on the IC is connected to which input or which output of which gate.

There are many digital ICs so we'll not show you the pinout of all of them here. However, for the practical exercises in this module, you'll only be working with six digital ICs. So the pinouts for these chips are shown below in Figure 33. The six ICs you'll be working with are:

- 7400 - Quad 2-input NAND gates
- 7402 - Quad 2-input NOR gates
- 7404 - Hex NOT gates
- 7408 - Quad 2-input AND gates
- 7432 - Quad 2-input OR gates
- 7486 - Quad 2-input XOR gates

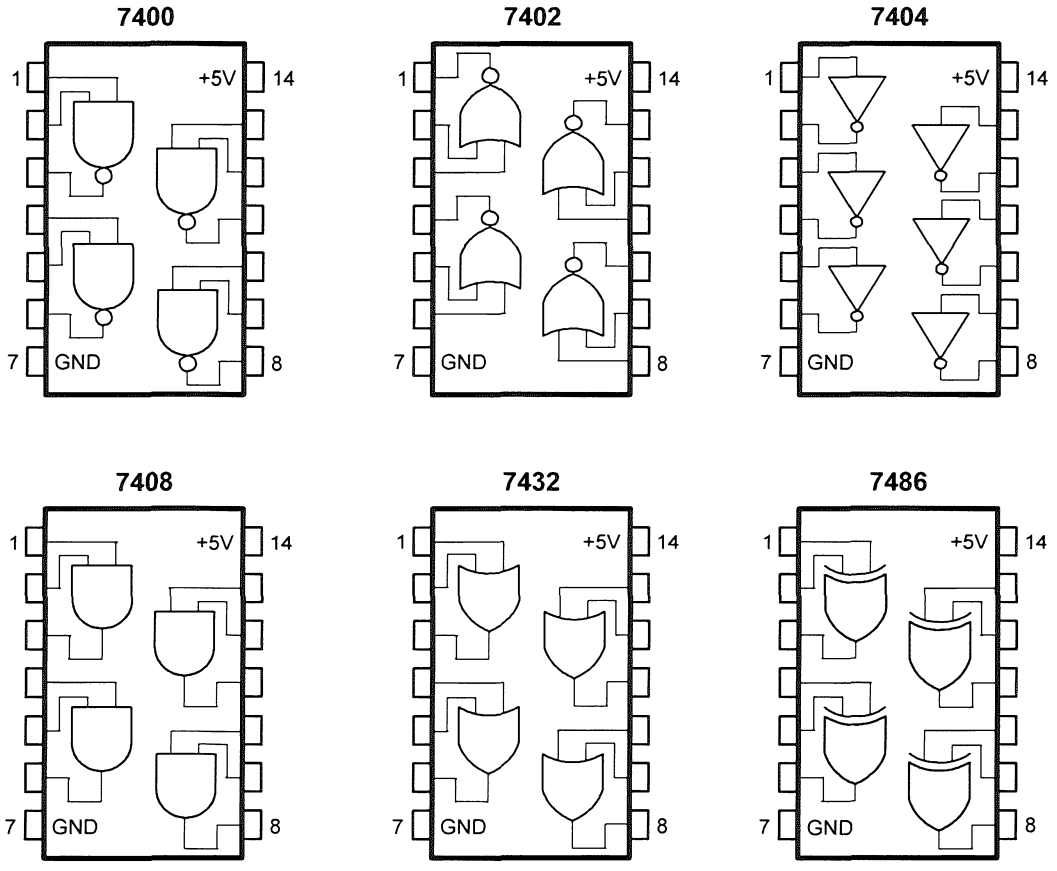


Figure 33

Skill practice 2

Practise testing and measuring digital ICs

This exercise is a practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 1¼ hours to complete this exercise.

Equipment

- Digital trainer
- DMM
- logic probe
- 7400*
- 7402*
- 7404*
- 7408*
- 7432*
- 7486*
- hook-up wire (terminated)
- IC extractor

* The part number of the ICs that you will be using may have letters after the 74 (eg 74LS00 or 74H00). These letters can be ignored for the purpose of this practical.

Remember:

Follow TAFE NSW WHS guidance at all times!

 Copyright: Not to be reproduced without permission

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

2. Gather the equipment needed for this exercise.

3. Set up supply rails (ie rows of +5V and ground connections) on the digital trainer's breadboard like you did for the Section 1 practical (see page 1-30).

4. Set the logic probe to the *TTL* and *PULSE* modes then use hook-up wire to connect its power leads to the appropriate supply rails.

5. Look up the pinout data for the **7408** (on page 2-26 or at the back of the book) and identify its power supply pins (+5V and ground). Record the pin numbers in Table 1 (to the right).

6. Select one of the logic gates inside the IC then identify and record the pin number of that gate's input pins and output pin.

7. Plug the IC into the breadboard.

8. Connect the IC's +5V pin to the breadboard's +5V rail and its GND pin to the ground rail.

Table 1

+5V pin #	
Gnd pin #	
Input-A pin #	
Input-B pin #	
Output (Z) pin #	

9. Connect both inputs of your chosen logic gate to toggle switches.

10. Set both toggle switches to the logic-0 position then turn the trainer on.

11. Use the logic probe to check that the logic levels on the input pins of the IC are logic-0 (this is just a check to make sure the digital trainer is working and that you've correctly wired the switches to the IC).

12. Measure the logic level on the output pin of the logic gate using the logic probe. Record your measurement in Table 2 (on the next page).

13. Measure and record the voltage on the output pin of the logic gate using the DMM.

14. Repeat steps 10 to 13 for the other three possible input combinations (that is, 01, 10 and 11).

Table 2

Inputs		Output	
Logic levels		Logic level	Voltage
0	0		
0	1		
1	0		
1	1		

Question 1
According to the IC's pinout data, what type of logic gate is inside the 7408?

Question 2
Compare your results in Table 2 with the truth table for this logic gate in these notes. Is the operation of the gate correct?

The teacher needs to check your work at this point...

15. Disconnect the wire from the IC's +5V pin to the +5V rail.

16. Measure the logic level on the output of the logic gate using the logic probe for the four possible input combinations and compare them with your results in Table 2 (above).

Question 3
Is it necessary for power to be connected to a digital IC for it to work? Explain why you think this is so.

17. Reconnect the wire from the IC's +5V pin to the +5V rail.

18. Disconnect one of the gate's inputs from the toggle switch and connect the input to the ground rail instead. This will "hard-wire" the input to a logic-0.

19. Operate the toggle switch still connected to the other input and use the logic probe to observe what happens to the logic gate's output.

Question 4
Which of the following statements describes the gate's operation now?

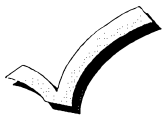
- ☐ The gate's output is always logic-0.
- ☐ The gate's output is always logic-1.
- ☐ The gate's output is the same as the logic level on the switch.
- ☐ The gate's output is the opposite of the logic level on the switch.

20. Disconnect the gate input that is connected to the ground rail and connect it to the +5V rail. This will hard-wire the input to a logic-1.

21. Operate the toggle switch and use the logic probe to observe what happens to the logic gate's output.

Question 5
Which of the following statements describes the gate's operation now?

- ☐ The gate's output is always logic-0.
- ☐ The gate's output is always logic-1.
- ☐ The gate's output is the same as the logic level on the switch.
- ☐ The gate's output is the opposite of the logic level on the switch.



The teacher needs to check your work at this point...

 Copyright: Not to be reproduced without permission

22. Remove the 7408 from the breadboard.

23. Look up the pinout data for the **7432** (on page 2-26) and identify its power supply pins. Record these in Table 4 below.

24. Repeat steps 6 to 14 recording your results in Tables 4 and 5 below.

Table 4

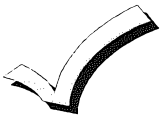
+5V pin #	
Gnd pin #	
Input-A pin #	
Input-B pin #	
Output (Z) pin #	

Table 5

Inputs		**Output**	
Logic levels		**Logic level**	**Voltage**
0	0		
0	1		
1	0		
1	1		

Question 6
According to the pinout data, what type of gate is inside the 7432?

Question 7
Compare your results in Table 5 with the truth table for this logic gate in the notes. Is the operation of the gate correct?



The teacher needs to check your work at this point...

25. Disconnect one of the gate's inputs from the toggle switch and hard-wire it to a logic-0.

26. Operate the toggle switch still connected to the other input and use the logic probe to observe what happens to the logic gate's output.

Question 8
Which of the following statements describes the gate's operation now?

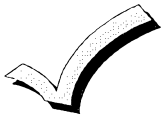
- ☐ The gate's output is always logic-0.
- ☐ The gate's output is always logic-1.
- ☐ The gate's output is the same as the logic level on the switch.
- ☐ The gate's output is the opposite of the logic level on the switch.

27. Hard-wire the same input to a logic-1 instead of a logic-0.

28. Operate the toggle switch and use the logic probe to observe what happens to the logic gate's output.

Question 9
Which of the following statements describes the gate's operation now?

- ☐ The gate's output is always logic-0.
- ☐ The gate's output is always logic-1.
- ☐ The gate's output is the same as the logic level on the switch.
- ☐ The gate's output is the opposite of the logic level on the switch.



The teacher needs to check your work at this point...

 Copyright: Not to be reproduced without permission

29. Remove the 7432 from the breadboard.

30. Look up the pinout data for the **7404** (on page 2-26) and identify its power supply pins. Record these in Table 6 below.

31. Repeat steps 6 to 14 recording your results in Tables 6 and 7 below.

Table 6

+5V pin #	
Gnd pin #	
Input pin #	
Output (Z) pin #	

Table 7

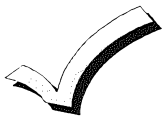
Input	**Output**	
Logic levels	**Logic level**	**Voltage**
0		
1		

Question 10
According to the pinout data, what type of gate is inside the 7404?

_____

Question 11
Compare your results in Table 7 with the truth table for this logic gate in the notes. Is the operation of the gate correct?

_____



The teacher needs to check your work at this point...

32. Remove the 7404 from the breadboard.

33. Look up the pinout data for the **7400** (on page 2-26) and identify its power supply pins. Record these in Table 8 below.

34. Repeat steps 6 to 14 recording your results in Tables 8 and 9 below.

Table 8

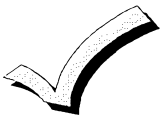
+5V pin #	
Gnd pin #	
Input-A pin #	
Input-B pin #	
Output (Z) pin #	

Table 9

Inputs		**Output**	
Logic levels		**Logic level**	**Voltage**
0	0		
0	1		
1	0		
1	1		

Question 12
According to the pinout data, what type of gate is inside the 7400?

Question 13
Compare your results in Table 9 with the truth table for this logic gate in the notes. Is the operation of the gate correct?



The teacher needs to check your work at this point...

35. Disconnect one of the gate's inputs from the toggle switch and hard-wire it to a logic-0.

36. Operate the toggle switch still connected to the other input and use the logic probe to observe what happens to the logic gate's output.

Question 14
Which of the following statements describes the gate's operation now?

- ☐ The gate's output is always logic-0.
- ☐ The gate's output is always logic-1.
- ☐ The gate's output is the same as the logic level on the switch.
- ☐ The gate's output is the opposite of the logic level on the switch.

37. Hard-wire the same input to a logic-1 instead of a logic-0.

38. Operate the toggle switch and use the logic probe to observe what happens to the logic gate's output.

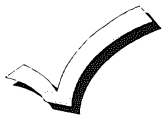
Question 15
Which of the following statements describes the gate's operation now?

- ☐ The gate's output is always logic-0.
- ☐ The gate's output is always logic-1.
- ☐ The gate's output is the same as the logic level on the switch.
- ☐ The gate's output is the opposite of the logic level on the switch.

Question 16
What other gate does the NAND act like when one of inputs is hard-wired to a logic-1?

_____

_____



The teacher needs to check your work at this point...

39. Remove the 7400 from the breadboard.

40. Look up the pinout data for the **7402** (on page 2-26) and identify its power supply pins. Record these in Table 10 below.

41. Repeat steps 6 to 14 recording your results in Tables 10 and 11 below.

Table 10

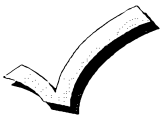
+5V pin #	
Gnd pin #	
Input-A pin #	
Input-B pin #	
Output (Z) pin #	

Table 11

Inputs		**Output**	
Logic levels		**Logic level**	**Voltage**
0	0		
0	1		
1	0		
1	1		

Question 17
According to the pinout data, what type of gate is inside the 7402?

Question 18
Compare your results in Table 11 with the truth table for this logic gate in the notes. Is the operation of the gate correct?



The teacher needs to check your work at this point...

42. Disconnect one of the gate's inputs from the toggle switch and hard-wire it to a logic-0.

43. Operate the toggle switch still connected to the other input and use the logic probe to observe what happens to the logic gate's output.

Question 19
Which of the following statements describes the gate's operation now?

- ☐ The gate's output is always logic-0.
- ☐ The gate's output is always logic-1.
- ☐ The gate's output is the same as the logic level on the switch.
- ☐ The gate's output is the opposite of the logic level on the switch.

44. Hard-wire the same input to a logic-1 instead of a logic-0.

45. Operate the toggle switch and use the logic probe to observe what happens to the logic gate's output.

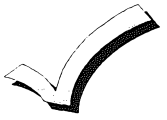
Question 20
Which of the following statements describes the gate's operation now?

- ☐ The gate's output is always logic-0.
- ☐ The gate's output is always logic-1.
- ☐ The gate's output is the same as the logic level on the switch.
- ☐ The gate's output is the opposite of the logic level on the switch.

Question 21
What other gate does the NOR act like when one of inputs is hard-wired to a logic-0?

_____

_____



The teacher needs to check your work at this point...

46. Remove the 7402 from the breadboard.

47. Look up the pinout data for the **7486** (on page 2-26) and identify its power supply pins. Record these in Table 12 below.

48. Repeat steps 6 to 14 recording your results in Tables 12 and 13 below.

Table 12

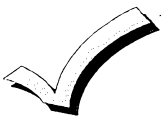
+5V pin #	
Gnd pin #	
Input-A pin #	
Input-B pin #	
Output (Z) pin #	

Table 13

Inputs		**Output**	
Logic levels		**Logic level**	**Voltage**
0	0		
0	1		
1	0		
1	1		

Question 22
According to the pinout data, what type of gate is inside the 7486?

Question 23
Compare your results in Table 13 with the truth table for this logic gate in the notes. Is the operation of the gate correct?



The teacher needs to check your work at this point...

Copyright: Not to be reproduced without permission

Student notes

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. Circle the NOR gates in Figure 1 below.

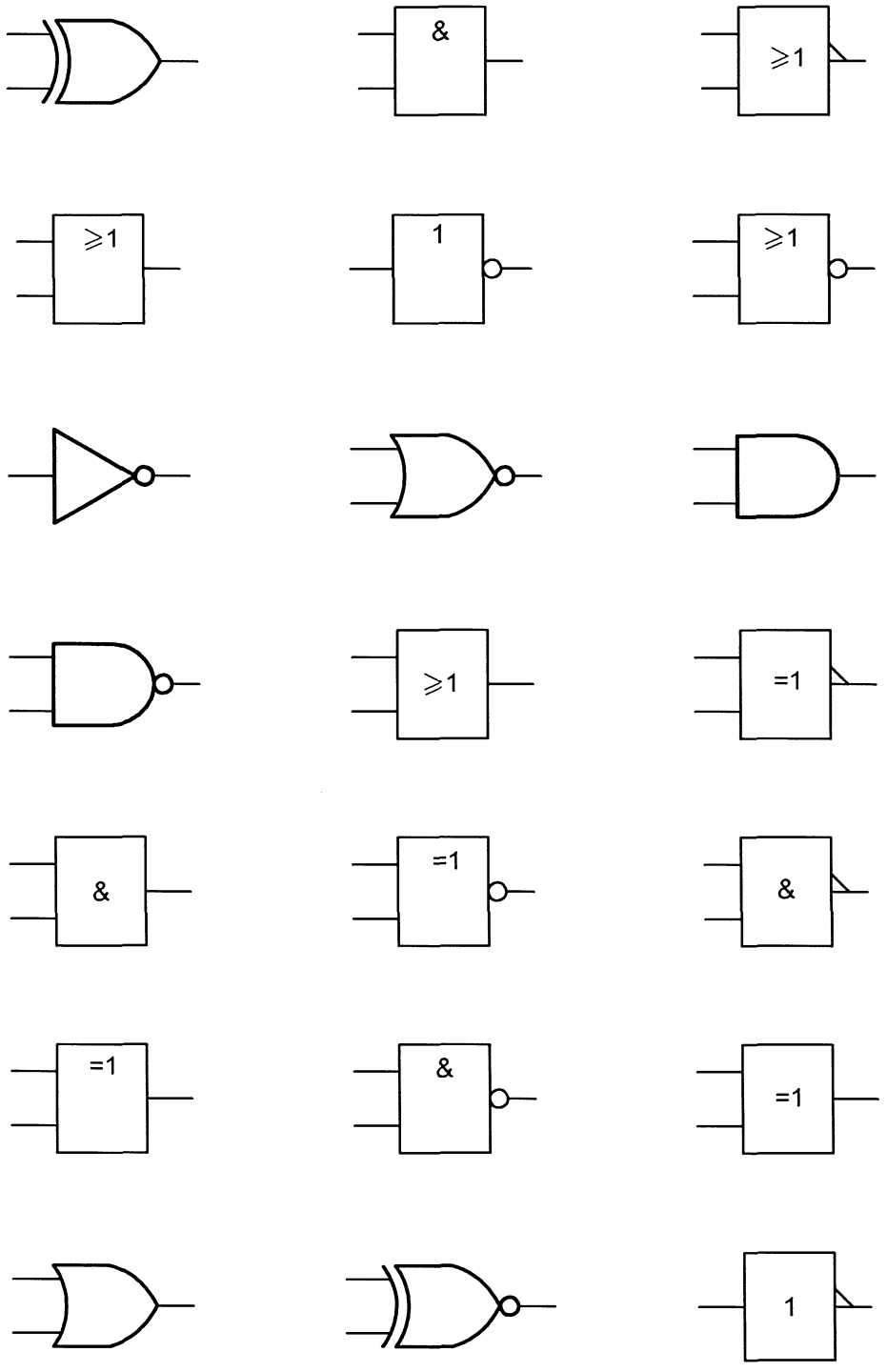


Figure 1

2. Circle the XOR gates in Figure 2 below.

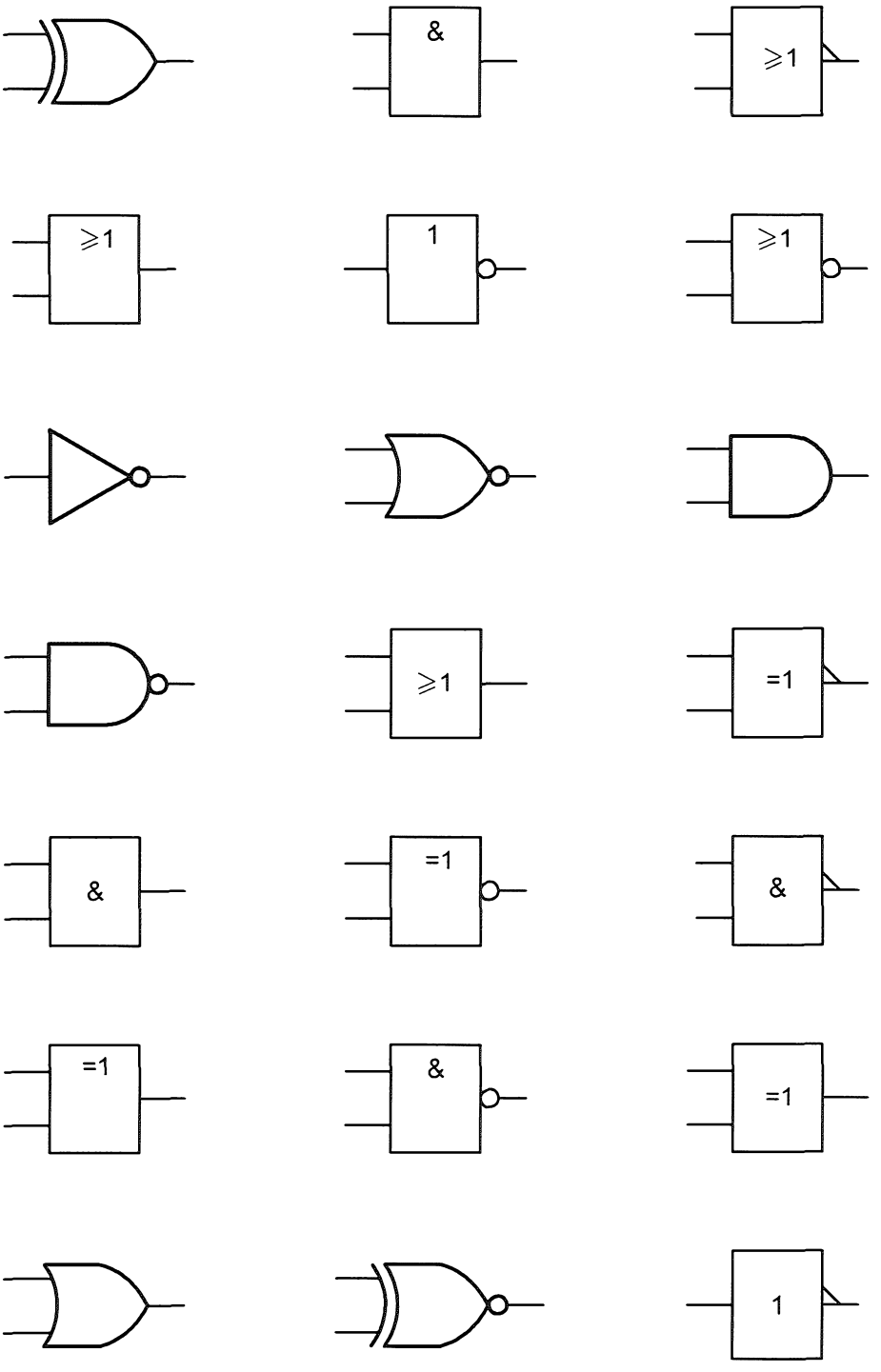


Figure 2

3. Circle the AND gates in Figure 3 below.

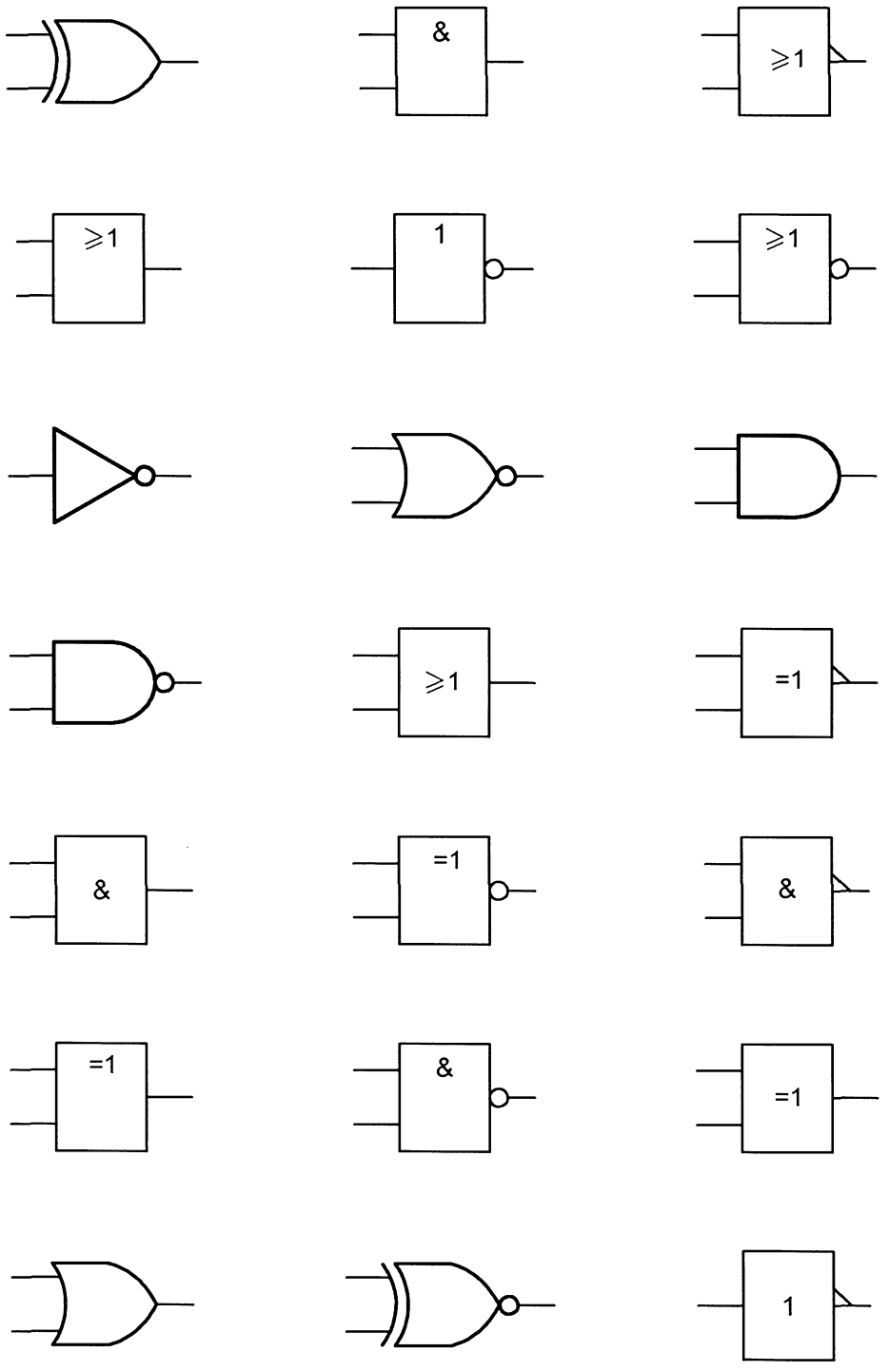


Figure 3

 Copyright: Not to be reproduced without permission

4. Label the inputs and outputs of the gate in Figure 4 below using letters of the alphabet.

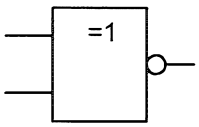


Figure 4

To help you answer questions 5 to 17, draw the truth table for the seven gates (a maximum of two inputs only) in the space provided directly below

5. Fill in the missing word in the following sentences:

   a) The output of a NAND gate will be _____ when both input-A and input-B are true. For all other input conditions, the output will be _____ .

   b) The output of an OR gate will be true if either input-A or input-B is _____ . If input-A and input-B are _____ , the output will be false.

   c) The output of an XOR gate will be _____ when input-A and input-B are the same. For all other input conditions, the output will be _____ .

   d) The output of the _____ gate will be true when the input is false. If the input is true, the output will be false.

6. Which of the following pairs of gates operate in the opposite way to each other? **Note:** There's more than one correct answer.

- ☐ AND and NAND
- ☐ OR and NOR
- ☐ XOR and XNOR
- ☐ OT and NOT

7. Which of the following gates will have a logic-0 on the output when both inputs are logic-0?

- ☐ OR
- ☐ AND
- ☐ NOR
- ☐ NAND
- ☐ XOR
- ☐ XNOR

8. Which of the following gates will have a logic-0 on the output when one input is a logic-0 and the other input is a logic-1?

- ☐ OR
- ☐ AND
- ☐ NOR
- ☐ NAND
- ☐ XOR
- ☐ XNOR

9. Which of the following gates will have a logic-0 on the output when both inputs are logic-1?

- ☐ OR
- ☐ AND
- ☐ NOR
- ☐ NAND
- ☐ XOR
- ☐ XNOR

10. Which gate will have a logic level on its output that is the same as the logic level on input-A if input B is hard-wired to a logic-1?

- ☐ OR
- ☐ AND
- ☐ NOR
- ☐ NAND

11. Which gate will have a logic level on its output that is the opposite of the logic level on input-A if input B is hard-wired to a logic-1?

- ☐ OR
- ☐ AND
- ☐ NOR
- ☐ NAND

12. Which gate will have a logic level on its output that is the same as the logic level on input-A if input B is hard-wired to a logic-0?

- ☐ OR
- ☐ AND
- ☐ NOR
- ☐ NAND

13. Which gate will have a logic level on its output that is the opposite of the logic level on input-A if input B is hard-wired to a logic-0?

- ☐ OR
- ☐ AND
- ☐ NOR
- ☐ NAND

14. Which gate's output will permanently be logic-1 if one of its inputs is hard-wired to a logic-1?

- ☐ OR
- ☐ AND
- ☐ NOR
- ☐ NAND

15. Which gate's output will permanently be logic-0 if one of its inputs is hard-wired to a logic-1?

- ☐ OR
- ☐ AND
- ☐ NOR
- ☐ NAND

16. Which gate's output will permanently be logic-0 if one of its inputs is hard-wired to a logic-0?

- ☐ OR
- ☐ AND
- ☐ NOR
- ☐ NAND

17. Which gate's output will permanently be logic-1 if one of its inputs is hard-wired to a logic-0?

- ☐ OR
- ☐ AND
- ☐ NOR
- ☐ NAND

Copyright: Not to be reproduced without permission

Questions 18 to 21 refer to Figure 5 below

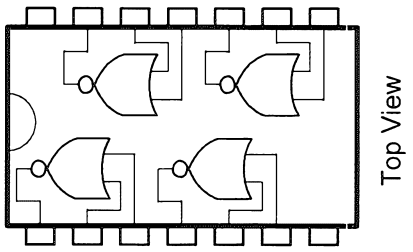


Figure 5

18. Identify the pin numbers on the IC (write the pin's number next to each pin).

19. Which IC pins are inputs?

_____

20. Which IC pins are outputs?

_____

21. What is the purpose of the two pins not connected to a gate?

_____

_____

Section 3 Combinational logic

Purpose To develop your ability to fault-find combinational logic circuits.

Objectives At the end of this section you should be able to:

- Identify combinational logic circuits in schematic diagrams of digital electronics equipment
- Predict the logic level on the output of a combinational logic circuit (up to three inputs and three levels) for a given set of logic levels on the inputs
- Develop the complete truth table for a combinational logic circuit (up to three inputs and three levels)
- List and explain four reasons for using combinational logic
- Draw the circuits that implement the NOT function using a single NAND gate or a single NOR gate
- Explain how to identify a faulty node in a combinational logic circuit
- Explain how to identify which logic gate is the cause of a faulty node in a combinational logic circuit
- Use a DMM and logic probe to test and measure combinational logic circuits and verify whether circuit nodes are the correct value

Introduction

The last section introduced you to the seven logic gates that are the building blocks of most logic circuits. Not surprisingly, logic gates can be connected together to form logic circuits that can perform complex functions and this is what you'll find inside digital electronics equipment.

As a technician, you'll likely have to locate faults in logic circuits. Consequently, you'll need to understand the basic operation of connected logic gate circuits, recognise when they're faulty and be able to locate the fault. This section looks at these issues.

Combinational logic

Logic circuits are made by simply connecting the outputs of one or more gates to the inputs of other gates. Logic circuits fall into two broad categories: *combinational logic* circuits and *sequential logic* circuits. Examples of combinational logic circuits are shown in Figure 1 below.

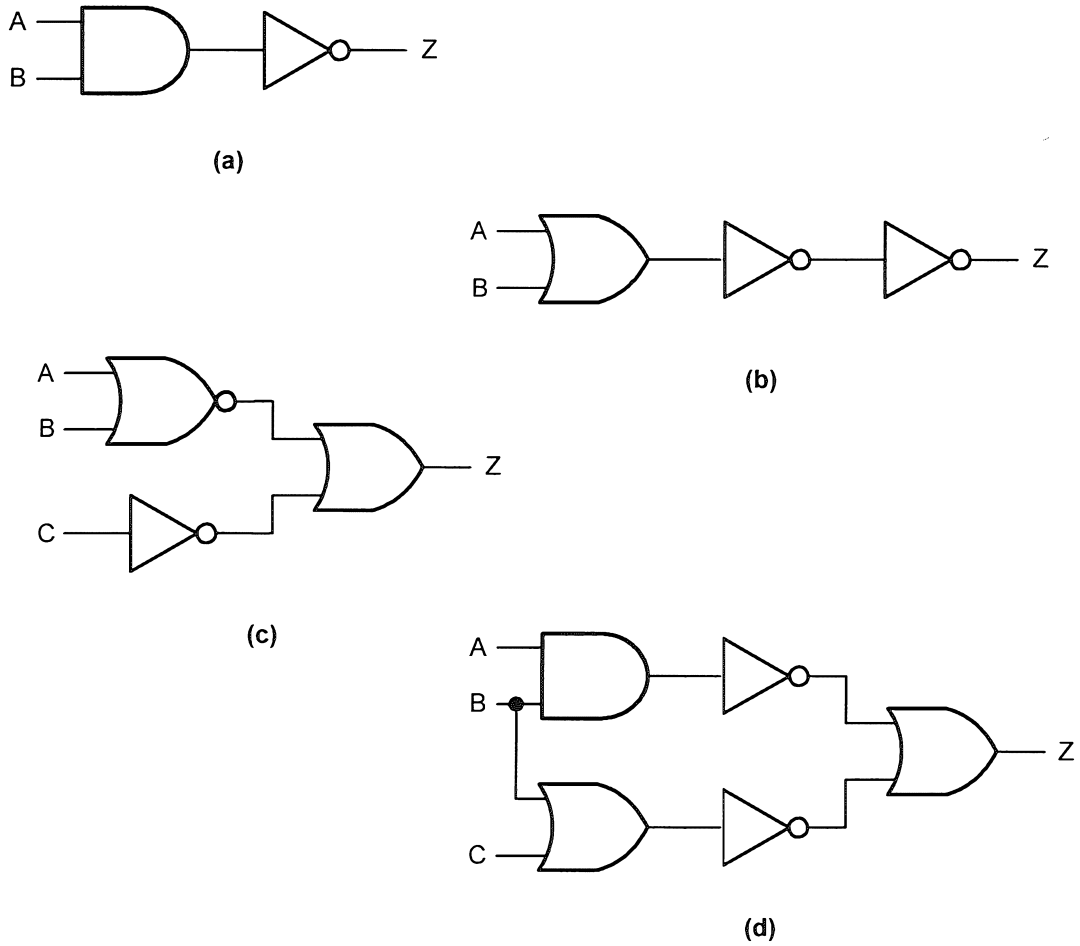


Figure 1 *Examples of combinational logic circuits*

If you look at the circuits in Figure 1 again, you'll notice that the inputs and outputs of individual gates are no-longer labelled. Only the inputs and outputs of the entire circuit are labelled and this is done in the same manner as for individual gates (that is, the inputs are labelled alphabetically starting from the letter A and the outputs are labelled counter-alphabetically starting from the letter Z).

It is common in textbooks to describe combinational logic circuits by the number of inputs to the entire circuit and/or the number of levels (or stages) in the circuit. For example, Figure 1a is a two input two level logic circuit, Figure 1b is a two input three level logic circuit, Figure 1c is a three input two level logic circuit and Figure 1d is a three input three level logic circuit.

Obviously, the more inputs to a combinational logic circuit, and the more levels it has, the more complex it is. For the purposes of this subject we'll mostly only consider the operation of circuits up to three inputs and three levels.

Combinational logic circuit operation

The operation of combinational logic circuits is surprisingly simple because individual gates always operate according to their truth table. So, for a given set of inputs to a combinational logic circuit, all you need do is determine the logic level on the outputs of the first level gates (that is, the ones connected to the circuit's inputs). You then use this information to determine the logic level on the outputs of the second level gates and so on until you've found the logic level on the circuit's output (or outputs).

This approach is demonstrated using the simple combinational logic circuit in Figure 2 below. Although the logic levels on the inputs to the circuit are known, it's impossible to predict the logic level on Z without first predicting the logic level on the output of the OR gate.

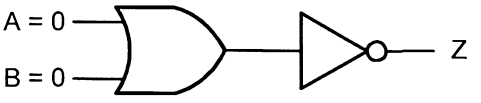


Figure 2 *A simple combinational logic circuit consisting of an OR gate and a NOT gate*

So, for the inputs to the OR gate shown, we can predict that its output will be logic-0 as shown in Figure 3 below.

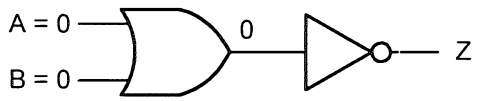


Figure 3 *With logic-0 on both of the OR gate's inputs, its output must also be logic-0*

And, unless there's a broken PCB track (which can happen), the NOT gate's input is a logic-0 also. This means that Z must be a logic-1 as shown in Figure 4 below.

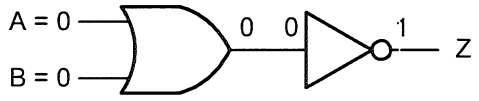


Figure 4 *With a logic-0 on the NOT gate's input, its output must be a logic-1*

This procedure can be used to determine the output of a combinational logic circuit regardless of the number of inputs and the number of levels of the circuit.

Practise predicting the logic level on the output of combinational logic circuits for yourself by analysing the following circuits given the input logic levels shown.

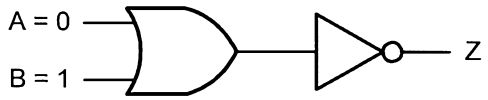


Figure 5

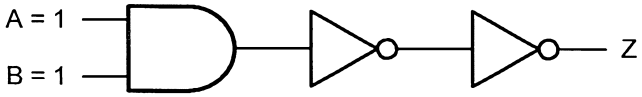


Figure 6

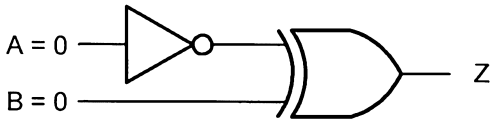


Figure 7

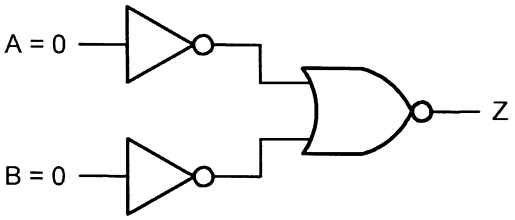


Figure 8

Copyright: Not to be reproduced without permission

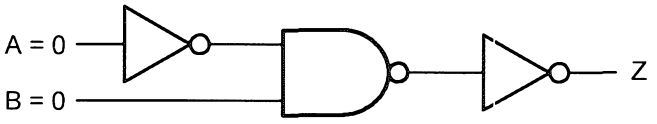


Figure 9

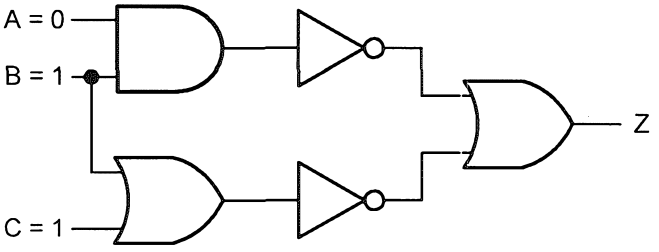


Figure 10

Developing a truth table for the output of a combinational logic circuit

Usefully, this technique of analysing combinational logic circuits to determine the output for a given set of inputs can be used to develop a complete truth table for the entire circuit. The procedure for this is as follows.

1. Identify the number of inputs to the circuit and write a truth table to suit, leaving the Z column blank.

   For example, if the circuit has two inputs the truth table would look like Table 1. If the circuit has three inputs the truth table would look like Table 2.

Table 1

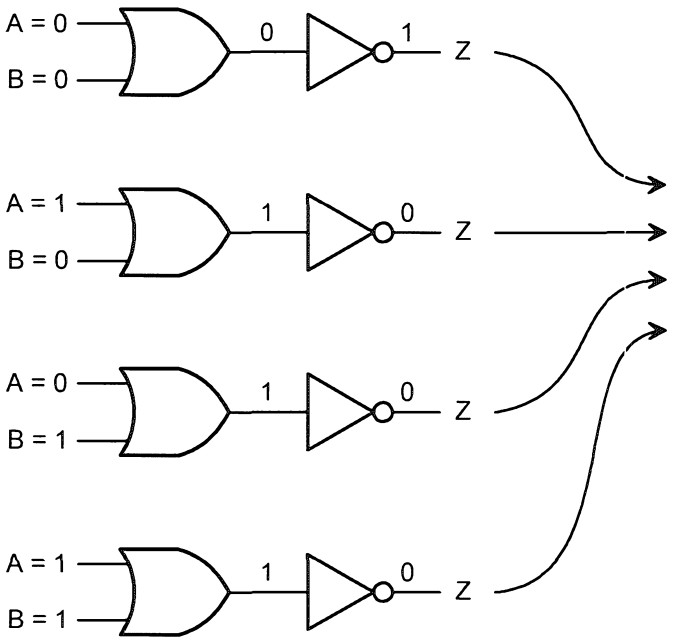
B	A	Z
0	0	
0	1	
1	0	
1	1	

Table 2

C	B	A	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

2. For each combination of input, work out the logic levels through the circuit until you have determined the logic level on Z. Then write this in the appropriate space in your truth table.

An example is shown for the simple two input two level combinational logic circuit in Figure 11 below.



B	A	Z
0	0	1
0	1	0
1	0	0
1	1	0

Figure 11

Practise this procedure for yourself by developing the truth table for the circuit in Figure 12.

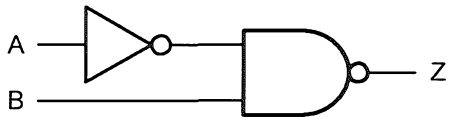


Figure 12

B	A	Z
0	0	
0	1	
1	0	
1	1	

For circuits with more than two inputs and/or more levels you might find it easier to develop the truth table output by output. An example of this is shown in Figure 13 below.

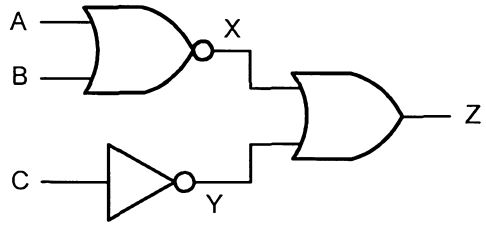


Figure 13

C	B	A	X	Y	Z
0	0	0	1	1	1
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	0	1	1
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	0	0

In this example, the outputs of the other gates have been labelled using the letters that precede the letter Z. The values for point X (which is just the NOR of A and B) is entered into the truth table followed by the values for point Y. After that, X and Y are used to solve for Z.

Practise this procedure for yourself by developing the truth table for the following two circuits.

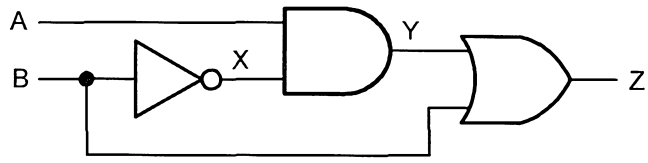


Figure 14

B	A	X	Y	Z
0	0			
0	1			
1	0			
1	1			

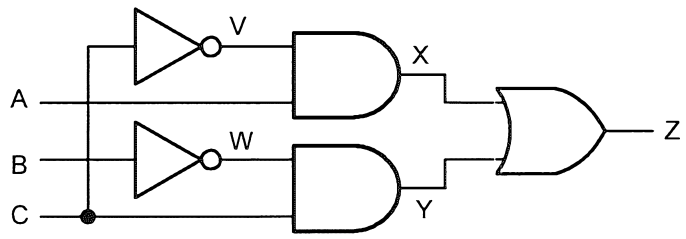


Figure 15

C	B	A	V	W	X	Y	Z
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

The reasons for using combinational logic circuits

There are four main reasons for using combinational logic circuits. One reason is to implement logic functions using other logic gates. Figures 16 to 19 show examples of this.

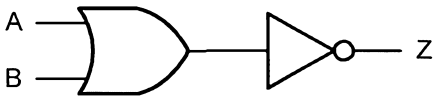


Figure 16 *A NOR gate made using an OR gate and a NOT gate*

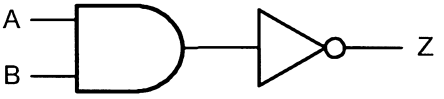


Figure 17 *A NAND gate made using an AND gate and a NOT gate*

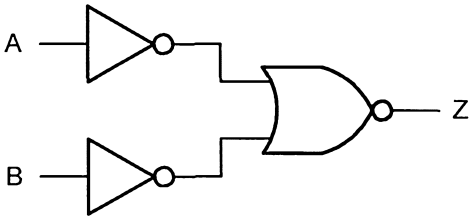


Figure 18 *An AND gate made using a NOR gate and two NOT gates*

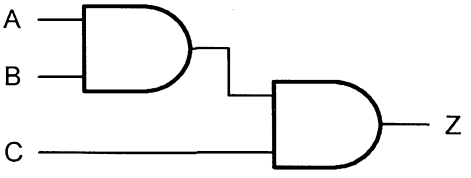
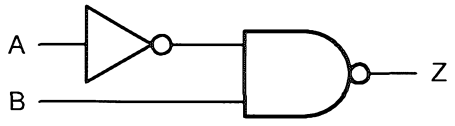


Figure 19 *A 3-input AND gate made using two 2-input AND gates*

Another reason for using combinational logic circuits is to implement truth tables not provided for by one of the seven logic gates. For example, the circuit in Figure 20 has the truth table shown to its right.



B	A	Z
0	0	1
0	1	1
1	0	0
1	1	1

Figure 20

A third reason for using combinational logic circuits is to implement digital electronics functions such as multiplexers, encoders and so on (you'll learn more about these devices later in this subject). A simple example of this is a circuit known as a binary half-adder (see Figure 21 below). This circuit can take two single bit binary numbers and give an output that is the sum of the two and, where applicable, produce a carry out.

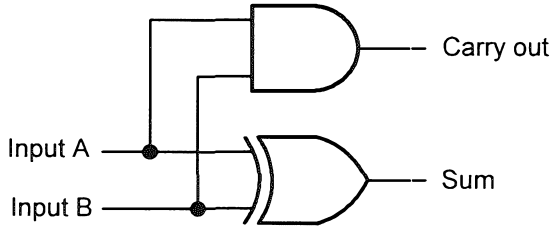


Figure 21 *A binary half-adder*

A fourth reason for using combinational logic (and probably the most important reason for using it until the advent of microcontrollers) is to control electrical, mechanical or electromechanical functions. Examples of this include: controlling a set of traffic lights, controlling a boom gate in a car park and controlling the paper-feed motors of a printer.

An example of a simple combinational logic circuit in a simple alarm system is shown in Figure 22 below.

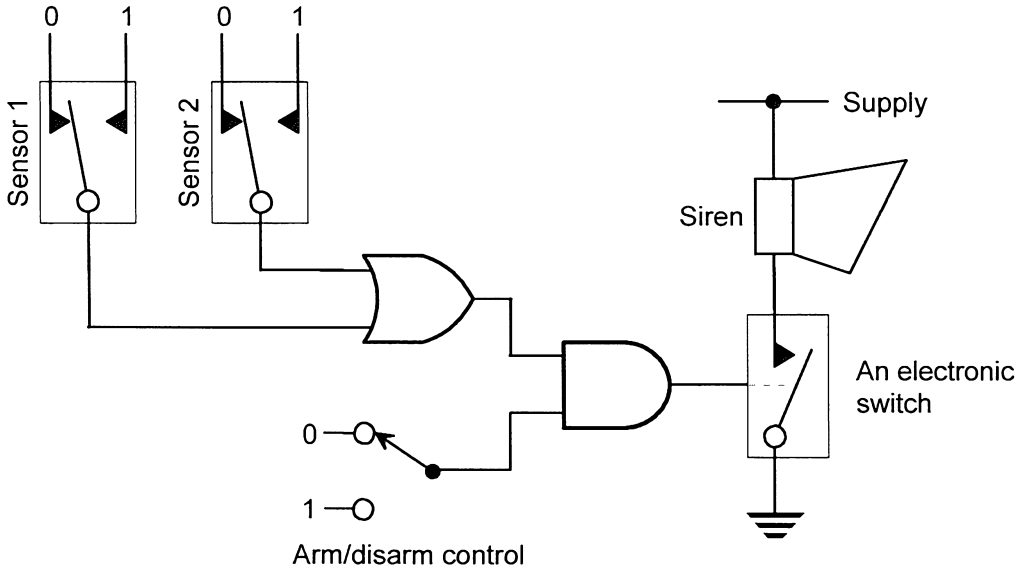


Figure 22 *A combinational logic circuit used in a simple alarm system*

As you can see, the OR gate is used so that two sensors can be connected to the circuit either of which can set-off the alarm. And, the AND gate is used so that the alarm can be armed and disarmed. Importantly, it would be impossible to implement this particular alarm system with just one logic gate.

NAND and NOR gates as NOT gates

As an interesting aside to this discussion on combinational logic, NAND gates and NOR gates can be wired to implement the NOT function without any additional components. To demonstrate this, let's start with the circuit in Figure 23 below.

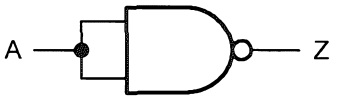


Figure 23 *The NAND gate wired to perform the same operation as a NOT gate*

As the NAND gate's inputs are hard-wired together, when input-A is logic-0 the gate's inputs are 00. According to the truth table of a NAND gate, an input of 00 gives a logic-1 on its output which is the opposite of the original logic level on A. And, when input-A is a logic-1 the gate's inputs are 11 which gives a logic-0 on its output (again, the opposite of A).

The NAND gate can also be wired to implement the NOT function using the circuit in Figure 24 below.

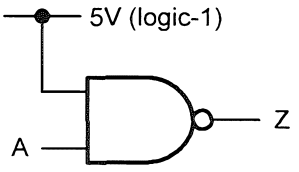


Figure 24 *Another method of wiring a NAND gate to perform the same operation as a NOT gate*

By hard-wiring one of the NAND gate's inputs to logic-1 in this way, when input-A is a logic-0 the gate's inputs are 01 and so its output is logic-1. And, when input-A is a logic-1, the gate's inputs are 11 and so its output is logic-0.

Figure 25 shows the first method of wiring a NOR gate to implement the NOT function.

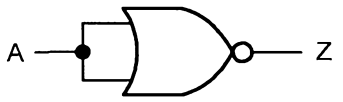


Figure 25 *The NOR gate wired to perform the same operation as a NOT gate*

As the NOR gate's inputs are hard-wired together, when input-A is logic-0 the gate's inputs are 00. According to the truth table of a NOR gate, an input of 00 gives a logic-1 on its output which is the opposite of the original logic level on A. And, when input-A is a logic-1 the gate's inputs are 11 which gives a logic-0 on its output (again, the opposite of A).

The NOR gate can also be wired to implement the NOT function using the circuit in Figure 26 below.

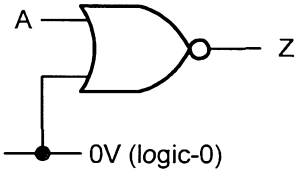


Figure 26 *Another method of wiring a NOR gate to perform the same operation as a NOT gate*

By hard-wiring one of the NOR gate's inputs to logic-0 in this way, when input-A is a logic-0 the gate's inputs are 00 and so its output is logic-1. And, when input-A is a logic-1, the gate's inputs are 01 and so its output is logic-0.

Fault-finding combinational logic circuits

There are two stages in the fault-finding process for combinational logic circuits. The first involves finding the faulty *node* in the circuit. (A node is a connection between a logic gate output and logic gate input or inputs.) The second stage involves identifying the gate that is causing the node to be faulty.

Finding the faulty node

To find a faulty node you must look for a logic level on the output of a logic gate that isn't what it should be. The procedure for this is as follows:

1. Measure the logic levels on the inputs to the entire circuit (see Figure 27).

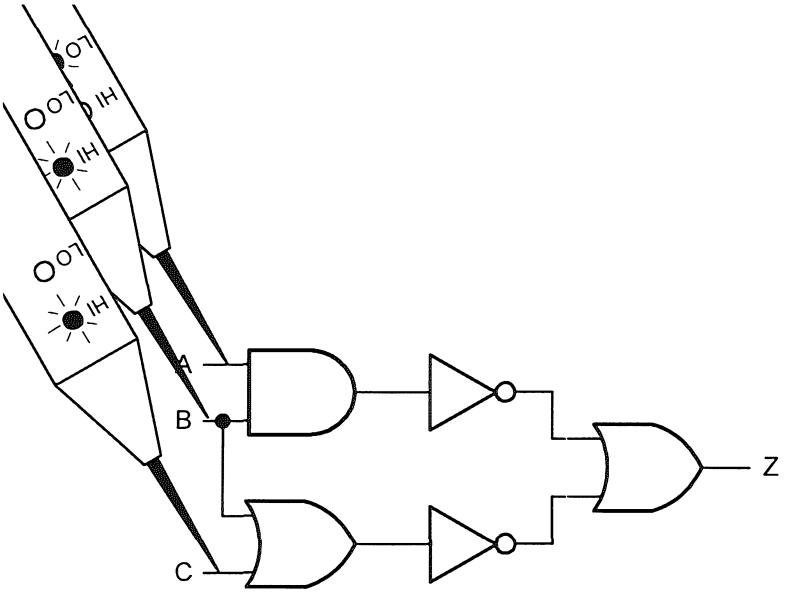


Figure 27

2. Once the inputs are known, use this information to predict and write down the theoretical logic level on the outputs of every gate in the circuit (see Figure 28).

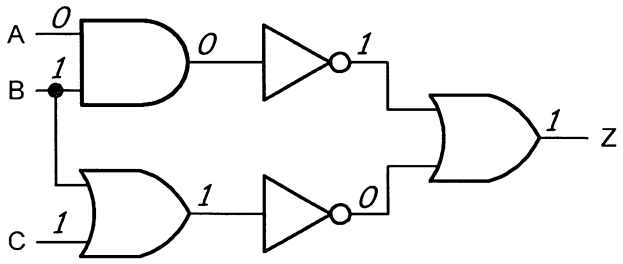


Figure 28

3. Measure the logic levels on the output of every gate and compare them to the predicted output. This is shown in Figures 29 to 32.

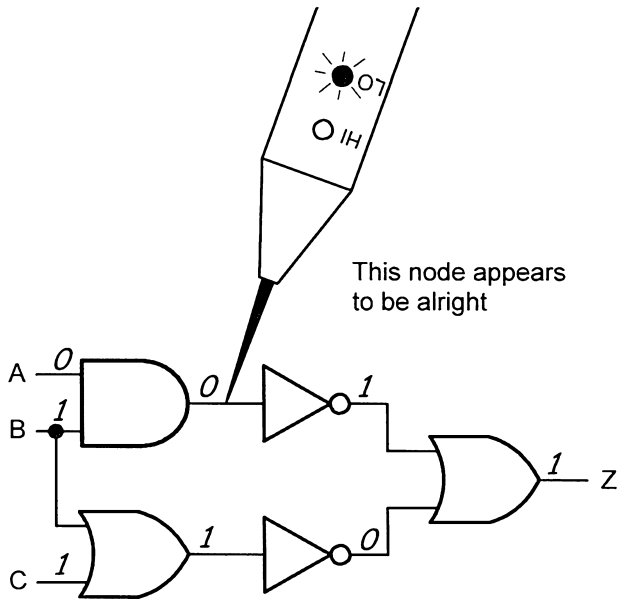


Figure 29

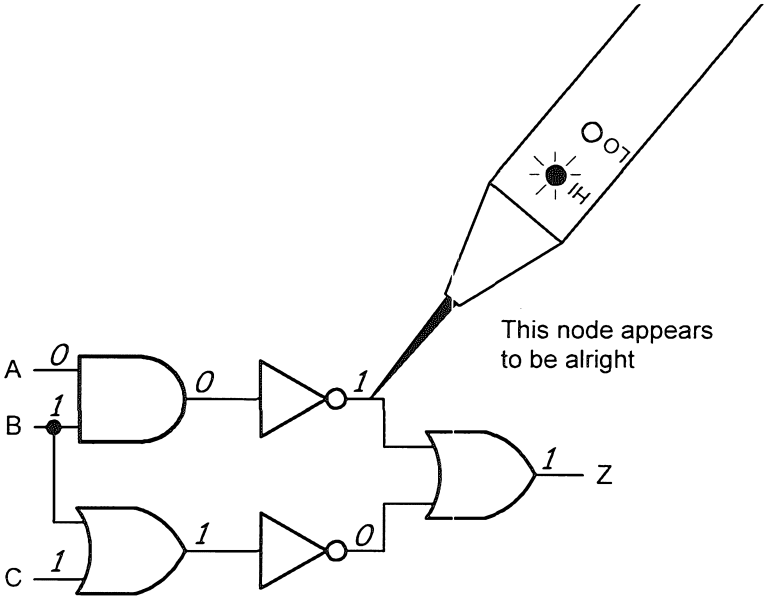


Figure 30

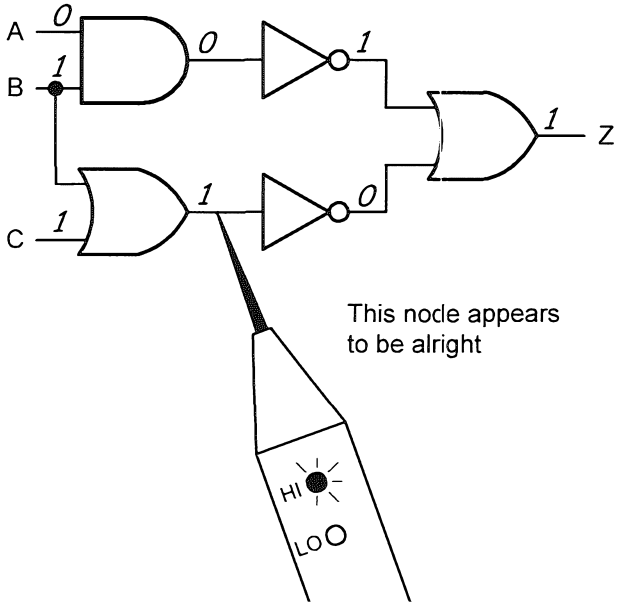


Figure 31

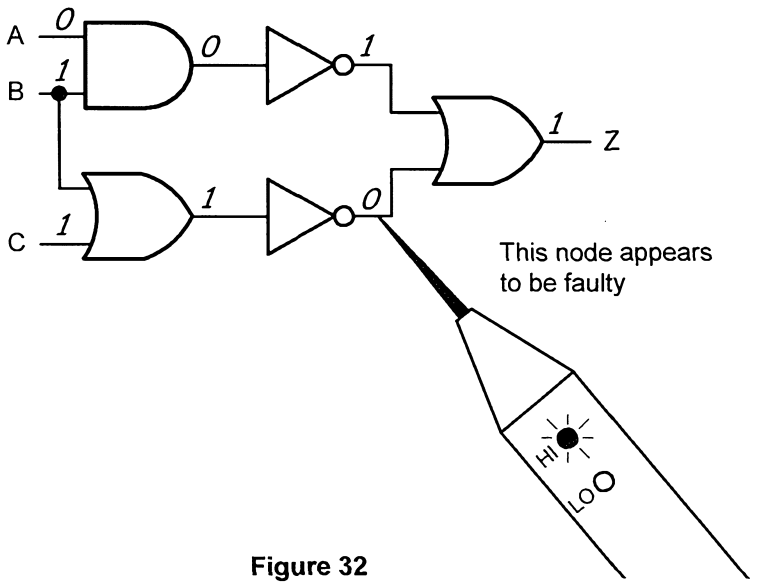


Figure 32

In this example, it appears that the connection between the lower NOT gate and the OR gate is a logic-1 when it should theoretically be a logic-0. If this node is always a logic-1, and never changes to a logic-0 regardless of the logic levels on the inputs to the circuit, then we say that the node is "stuck high". Similarly, if a node is always a logic-0, regardless of the logic levels on the inputs to the circuit, then we say that the node is "stuck low".

Care must be taken when fault-finding combinational logic circuits because the logic levels at the outputs of all the logic gates can appear to be correct even though the circuit is faulty. This can happen because a node might be stuck at a logic level that the input conditions to the circuit would produce anyway. For example, the faulty circuit above might not seem faulty when the inputs to the circuit are all logic-0 as shown in Figure 33 below.

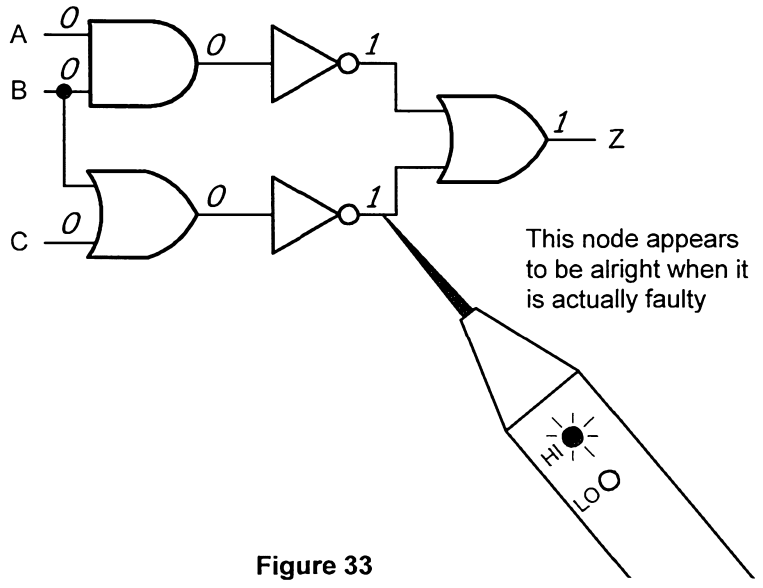


Figure 33

If a combinational logic circuit is suspected of being faulty then it is wise to test the circuit for all possible combinations of input logic levels. Having said that, sometimes it may not be possible to easily change the logic levels on the inputs to a combinational logic circuit. In which case, other strategies involving the combined use of a logic probe and a logic pulser can be used (logic pulsers are devices designed to momentarily override an existing logic level in logic circuits - see Appendix 1).

Finding the faulty gate

Once the faulty node is found, it is tempting to immediately blame the gate that has its output connected to the node (which, in the example we've been using, is the NOT gate). However, it is equally possible that the gate (or gates) that has an input connected to the node is the culprit (see Figure 34 below).

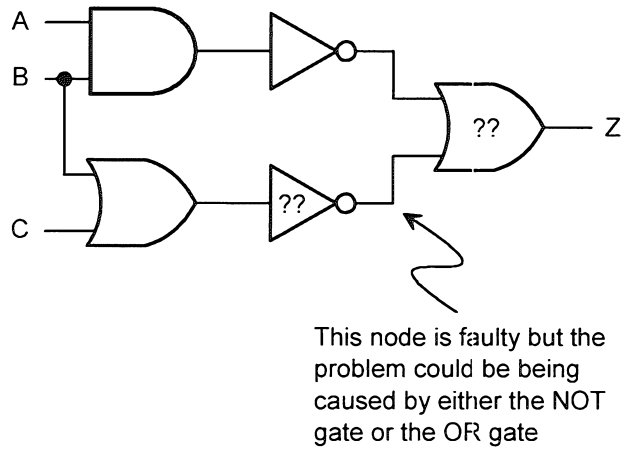


Figure 34

To explain, the simplicity of logic gate schematic symbols disguises the fact that they're actually a little circuit consisting of several transistors, resistors and diodes. So, when a logic gate is faulty, one or more of these tiny components are damaged. The effect of the this will depend on where inside the gate's circuit the damaged component is connected. The fault may simply cause a gate to misbehave. However, if the fault occurs at the gate's input, the problem can over-ride the driving gate's output and pull it up to logic-1 or down to logic-0.

To identify which gate is causing the faulty node, use the following procedure which is demonstrated using the circuit from the previous few pages.

1. Isolate the logic gate output connected to the faulty node from the circuit (see Figure 35 below).

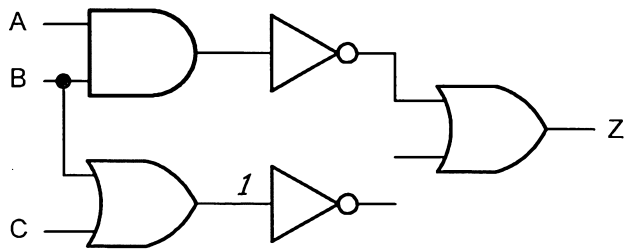


Figure 35

Isolating a logic gate output or input from the node involves making sure it is disconnected from the PCB track that it is normally connected to. This can be done in one of several ways but the most convenient method is to remove the IC then re-insert it with the pin of the output (or input) being bent outward rather than downward.

2. Check the logic level on the logic gate's output (not on the node). If the logic gate's output remains stuck then it is the faulty gate (see Figure 36).

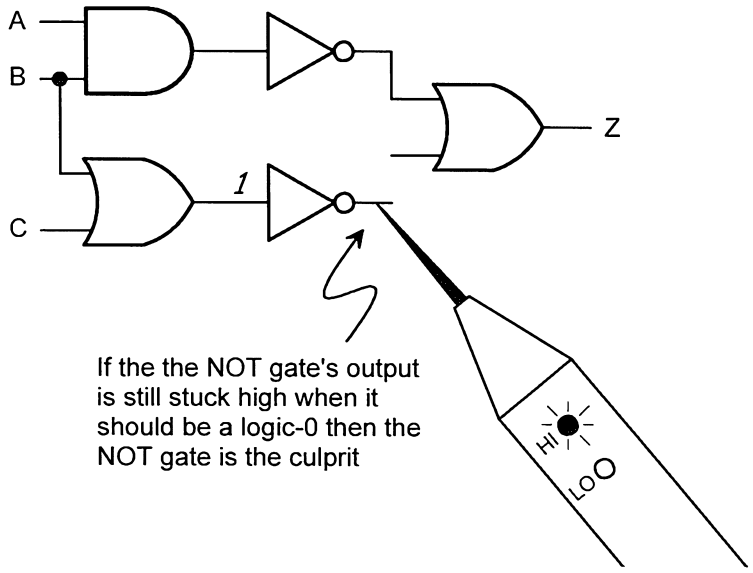


Figure 36

However, if the logic level on the logic gate's output is what it should be then the gate is probably ok and the problem is with the logic gate that has its input connected to the node (see Figure 37).

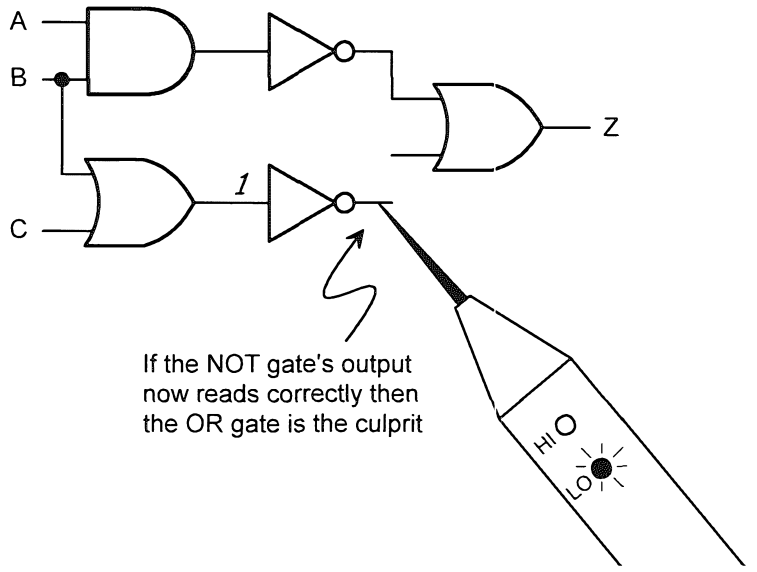


Figure 37

If there is more than one logic gate input connected to the node then more tests need to be made to identify which gate's input is causing the faulty node. This is done by reconnecting the logic gate's output to the circuit then isolating each logic gate input one at a time and testing the node (not the logic gate input). When the logic level on the node reads correctly, the gate with the isolated input is the culprit.

Skill practice 3

An introduction to testing and measuring combinational logic circuits

This exercise is a practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 1½ hours to complete this exercise.

Equipment

- Digital trainer
- DMM
- logic probe
- 7400*
- 7402*
- 7404*
- 7408*
- 7432*
- 7486*
- hook-up wire (terminated)
- IC extractor

* The part number of the ICs that you will be using may have letters after the 74 (for example, 74LS00 or 74H00). These letters can be ignored for the purpose of this practical.

Remember:

Follow TAFE NSW WHS guidance at all times!

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

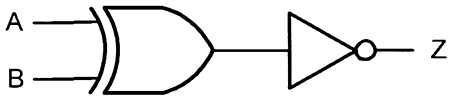
My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

2. Gather the equipment needed for this exercise.

3. For the circuit in Figure 1 below, use the pinout data on page 2-26 (or at the back of the book) to identify the ICs that contain the gates shown. Record the ICs' number in the space provided below.

4. Identify and record the power supply pins of both ICs.



IC:_____ IC:_____

Vcc pin:___ Vcc pin:___

Gnd pin:___ Gnd pin:___

Figure 1

5. Pick one of the XOR gates inside the 7486 to use to wire the circuit. Record the pin numbers of that gate's inputs and output in Figure 1 next to the input and output pins of the XOR gate.

6. Repeat step 5 for one of the NOT gates inside the 7404.

7. Set up +5V and ground supply rails on the digital trainer's breadboard like you did for the practical in Section 1 (see page 1-30).

8. Wire the circuit of Figure 1. **Before** you do, read the following notes.

   - Handle all ICs carefully - remember ESD can damage them.
   - Remember to connect power to both of the ICs.
   - Connect inputs A and B to their own toggle switch.
   - Remember that when plugging hook-up wire into the breadboard to make connections to the pins of the IC, use holes that are the furthest away from the IC's pin.

9. Set the logic probe to the *TTL* and *PULSE* modes then connect its power leads to the breadboard's power supply rails (+5V and GND).

10. Set both of the toggles switches to logic-0.

11. Measure the logic level on the output of the circuit (*Z*). Record this in Table 1 below.

12. Repeat steps 10 and 11 for the other three input combinations.

Table 1

Input B	Input A	Z
0	0	
0	1	
1	0	
1	1	

Question 1
From the results in Table 1, what logic gate has the same truth table as this circuit?

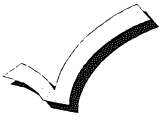
_____

Question 2
Briefly explain why the circuit has the same truth table as an XNOR gate.

_____

_____

_____



The teacher needs to check your work at this point...

13. If the teacher has checked your work, dismantle the circuit of Figure 1.

14. Identify the two ICs that you need to implement the circuit in Figure 2 below.

15. Identify the ICs' power supply pins and the input and output pins of the gates inside them that you plan to use. Write these pin numbers on the diagram in Figure 2.

16. Insert the two ICs into the breadboard and connect the power supply to them.

17. Wire the circuit of Figure 2 connecting inputs A and B to their own toggle switch.

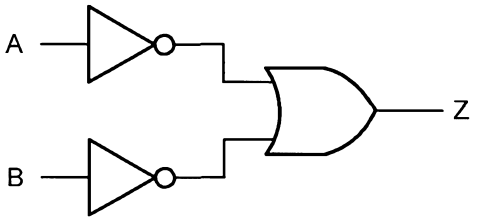


Figure 2

18. Set both of the toggles switches to logic-0.

19. Measure the logic level on the output of the circuit (Z). Record this in Table 2 below.

20. Repeat steps 18 and 19 for the other three input combinations.

Table 2

Input B	Input A	Z
0	0	
0	1	
1	0	
1	1	

Question 3
What logic gate has the same truth table as this circuit?

The teacher needs to check your work at this point...

21. If the teacher has checked your work, dismantle the circuit of Figure 2.

22. Repeat steps 14 to 16 for the circuit in Figure 3 below.

23. Wire the circuit of Figure 3 connecting inputs A and B to their own toggle switch.

24. Measure the logic level on the output of the circuit (Z) for all possible input combinations and record these in Table 3 below.

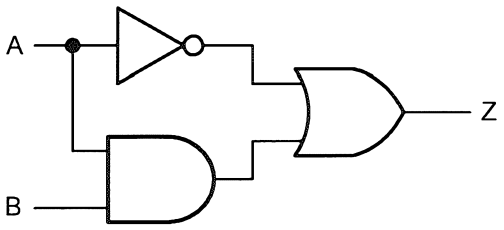
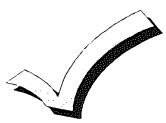


Figure 3

Table 3

Input B	Input A	Z
0	0	
0	1	
1	0	
1	1	

Question 4
Is there a logic gate with this truth table?



The teacher needs to check your work at this point...

25. If the teacher has checked your work, dismantle the circuit of Figure 3.

26. Repeat steps 14 to 16 for the circuit in Figure 4 below.

27. Wire the circuit of Figure 4 connecting inputs A and B to their own toggle switch.

28. Measure the logic level on the output of the circuit (Z) for all possible input combinations and record these in Table 4 below.

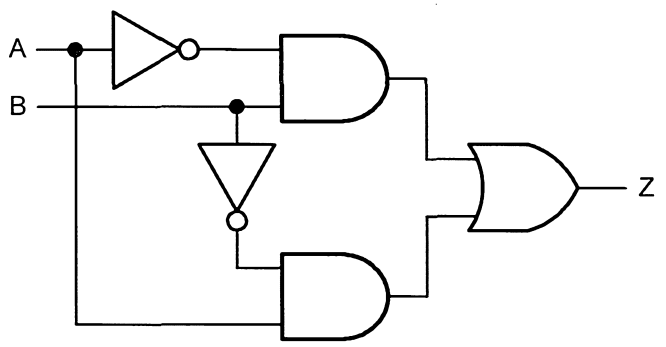
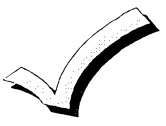


Figure 4

Table 4

Input B	Input A	Z
0	0	
0	1	
1	0	
1	1	



The teacher needs to check your work at this point...

29. If the teacher has checked your work, dismantle the circuit of Figure 4.

30. Repeat steps 14 to 16 for the circuit in Figure 5 below.

31. Wire the circuit of Figure 5 connecting inputs A, B and C to their own toggle switch.

32. Measure the logic level on the output of the circuit (Z) for all possible input combinations and record these in Table 5 below.

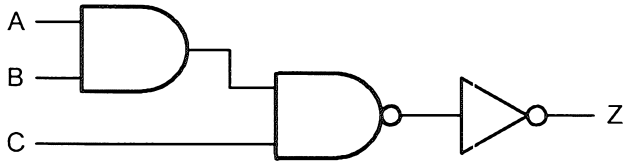
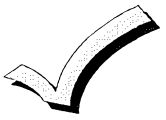


Figure 5

Table 5

Input C	Input B	Input A	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Question 5
What logic gate has the same truth table as this circuit?



The teacher needs to check your work at this point...

33. If the teacher has checked your work, dismantle the circuit of Figure 5.

34. Repeat steps 14 to 16 for the circuit in Figure 6 below.

35. Wire the circuit of Figure 6 connecting inputs A, B and C to their own toggle switch.

36. Measure the logic level on the output of the circuit (Z) for all possible input combinations and record these in Table 6 on the next page.

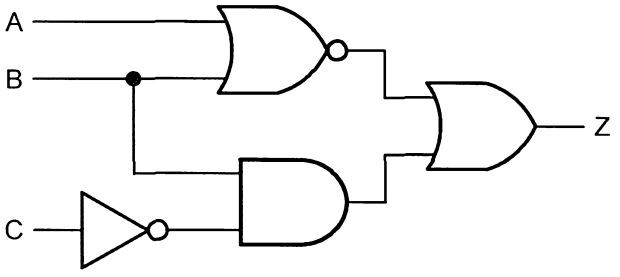
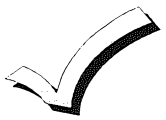


Figure 6

Table 6

Input C	Input B	Input A	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



The teacher needs to check your work at this point...

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. What are four reasons for using combinational logic?

2. Complete the truth table for the circuit in Figure 1 below.

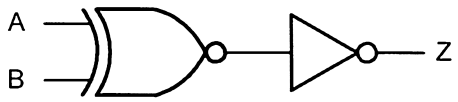


Figure 1

B	A	Z
0	0	
0	1	
1	0	
1	1	

3. What logic gate has the same truth table as this circuit?

4. Complete the truth table for the circuit in Figure 2 below.

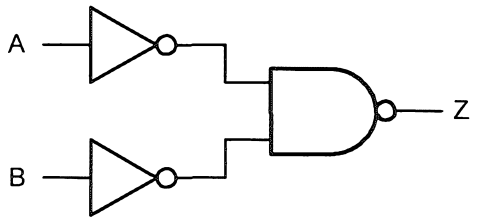


Figure 2

B	A	Z
0	0	
0	1	
1	0	
1	1	

5. What logic gate has the same truth table as this circuit?

_____

6. Complete the truth table for the circuit in Figure 3 below.

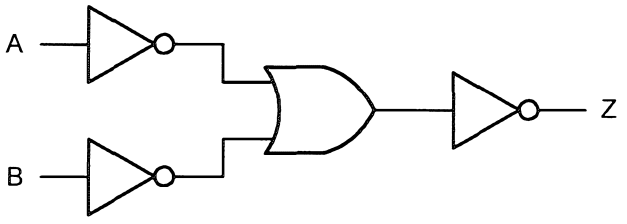


Figure 3

B	A	Z
0	0	
0	1	
1	0	
1	1	

7. What logic gate has the same truth table as this circuit?

_____

8. Complete the truth table for the circuit in Figure 4 below.

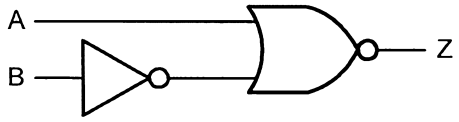


Figure 4

B	A	Z
0	0	
0	1	
1	0	
1	1	

9. What logic gate has the same truth table as this circuit?

_____

10. Complete the truth table for the circuit in Figure 5 below.

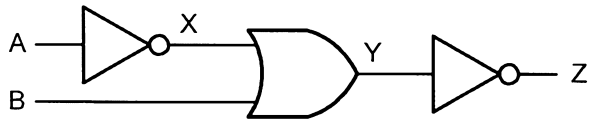


Figure 5

B	A	X	Y	Z
0	0			
0	1			
1	0			
1	1			

11. Complete the truth table for the circuit in Figure 6 below.

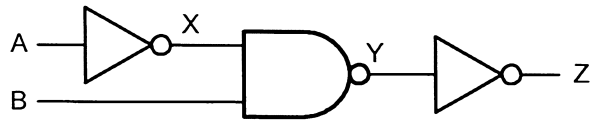


Figure 6

B	A	X	Y	Z
0	0			
0	1			
1	0			
1	1			

12. Complete the truth table for the circuit in Figure 7 below.

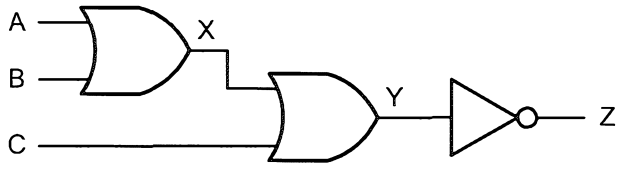


Figure 7

C	B	A	X	Y	Z
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

13. What logic gate has the same truth table as this circuit?

_____

14. Explain how you would identify which gate is faulty (4 or 5) in Figure 8 below if the if node Y was found to be faulty.

_____

_____

_____

_____

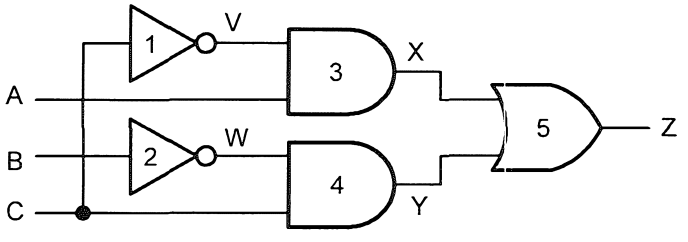


Figure 8

15. Complete the truth table for the circuit in Figure 9 below.

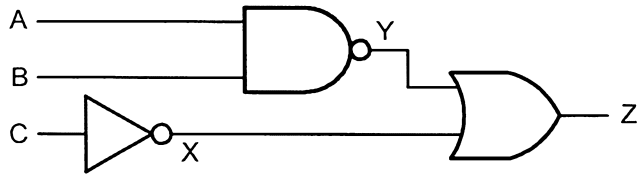


Figure 9

C	B	A	X	Y	Z
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

16. Which gate (or gates) in the circuit of Figure 10 is faulty.

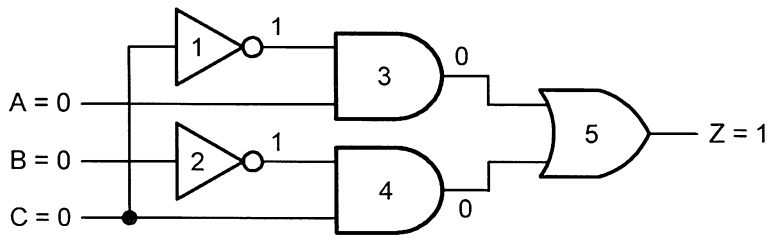


Figure 10

17. Which gates in the circuit of Figure 11 are possibly faulty on the information provided.

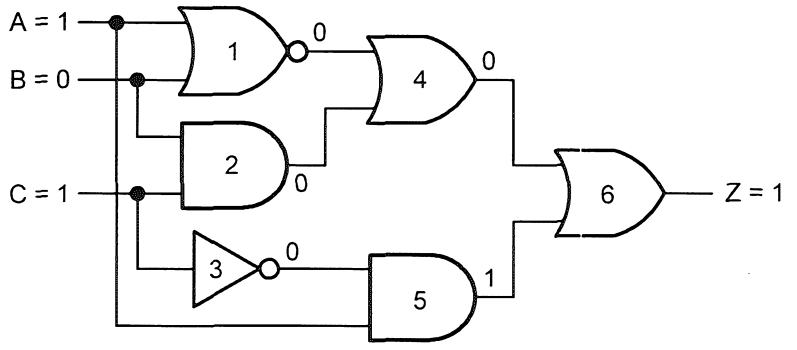


Figure 11

18. Which gates in the circuit of Figure 12 are possibly faulty on the information provided.

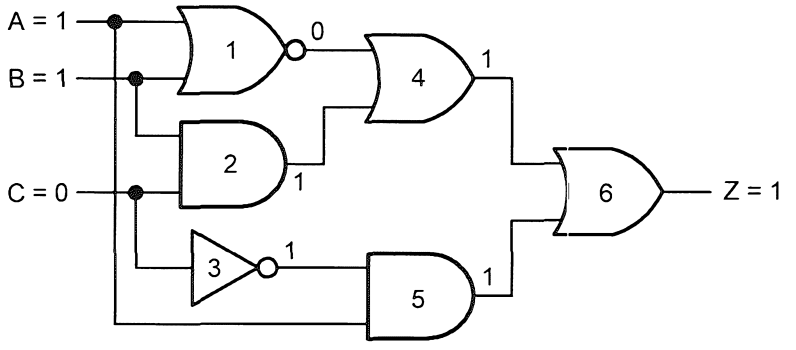


Figure 12

19. Which gates in the circuit of Figure 13 are possibly faulty on the information provided.

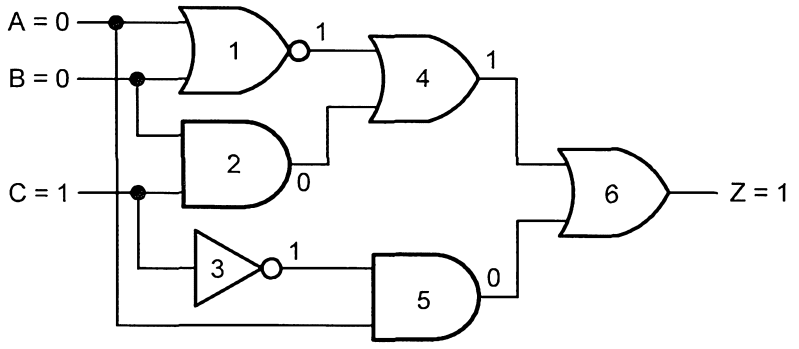


Figure 13

20. Which gates in the circuit of Figure 14 are possibly faulty on the information provided.

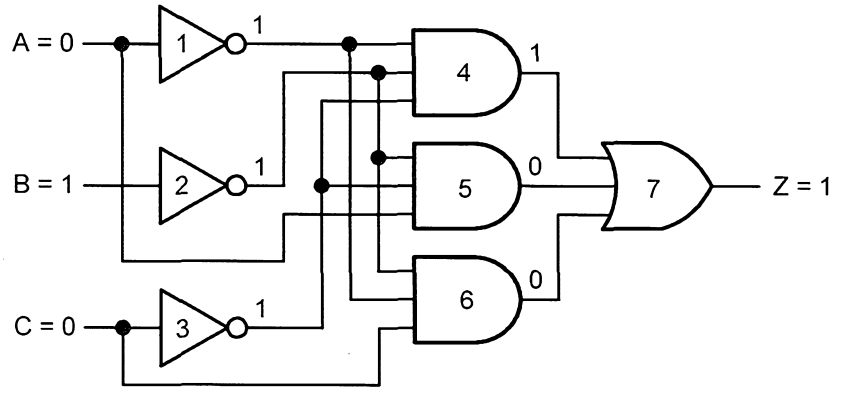


Figure 14

Copyright: Not to be reproduced without permission

21. Draw the two circuits that use a single 2-input NAND gate to implement the NOT function.

22. Draw the two circuits that use a single 2-input NOR gate to implement the NOT function.

23. Redraw the circuit in Figure 15 below so that the same logic function is implemented entirely from NOR gates.

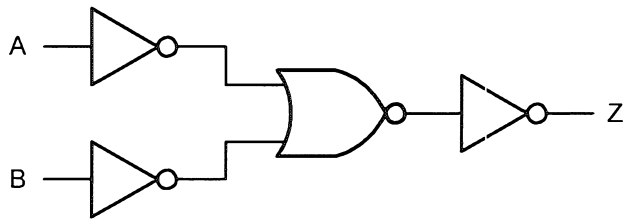


Figure 15

Section 4 Boolean algebra

Purpose To develop your ability to predict the logic level on the output of combinational logic circuits using Boolean algebra.

Objectives At the end of this section you should be able to:

- Describe the similarities and differences between Boolean algebra and conventional algebra
- Write the Boolean expression for each of the seven logic gates
- Develop the Boolean expression for combinational logic circuits (up to three inputs and three stages)
- Write a truth table from a sum-of-products Boolean expression
- State DeMorgan's theorems
- Simplify Boolean expressions using DeMorgan's theorems
- Analyse the operation of combinational logic circuits by writing the Boolean expression for its output, simplifying the expression to the sum-of-products form and writing the truth table
- Use a DMM and logic probe to test and measure combinational logic circuits and verify their operation

Introduction

The last section introduced you to combinational logic circuits, how they work and some of the reasons for using them. This section continues to look at combinational logic circuits by teaching you how to analyse them using Boolean algebra. Although Boolean algebra is more of a design tool, it is also used to communicate the operation of circuits and so technicians need to know a little about it.

Boolean algebra

Boolean algebra is a branch of mathematics developed by an English mathematician called George Boole (1815 to 1864) as part of his life long interest in logic. Boolean algebra works with the binary number system and the logical functions of AND, OR, NAND, NOR and NOT. As such, it is perfect for analysing the operation of combinational logic circuits.

There are several parts to Boolean algebra that you need to know about before we look at how it can be used to analyse logic circuits. These include: the *variables*, the *Boolean signs* and *Boolean expressions*.

Variables

Recall that a variable in conventional algebra is a letter that is used represent any number from $-\infty$ to $+\infty$. For example, the letters I, E and R are variables in Ohm's Law ($I = \frac{E}{R}$). Boolean algebra is similar to conventional algebra because it also uses letters (usually upper case) to represent numbers. However, there is one significant difference. Variables in Boolean algebra can only represent either zero or one.

Boolean signs

Boolean algebra borrows two of the signs used in conventional algebra: the dot ($\cdot$) (which is usually used in conventional algebra as a short-hand method of denoting multiplication) and the *plus* symbol (+). In Boolean algebra the dot ($\cdot$) represents the AND function and the plus sign (+) represents the OR function.

Boolean algebra also has its own sign which is a horizontal line written above a variable or groups of variables (for example, $\overline{A}$ or $\overline{A \cdot B}$). The line is used to denote the NOT function.

These three signs can be combined with variables to represent the operation of any logic function and, by extension, any logic gate.

Boolean expressions

An *expression* in conventional algebra is a statement that describes the relationship between one or more variables and one or more other variables. An example of a simple conventional algebraic expression is $c = a + b$. This expression states that the variable c is equal to the sum of the numerical values of a and b. (So, if a is the number 3 and b is the number 5, then c must be the number 8.)

A Boolean expression is similar in that it describes the relationship between a variable and one or more other variables. An example of a simple Boolean expression is $Z = A + B$.

At first, this Boolean expression looks like a conventional algebraic expression. However, remember that the variables in Boolean algebra can only be zero or one and the signs represent logical operations not mathematical operations.

The Boolean expressions for logic gates

As logic gates are circuits that implement the seven logic functions, it makes sense that their operation can be described using Boolean expressions. The Boolean expressions for the AND, OR and NOT gates are shown in Figure 1 below.

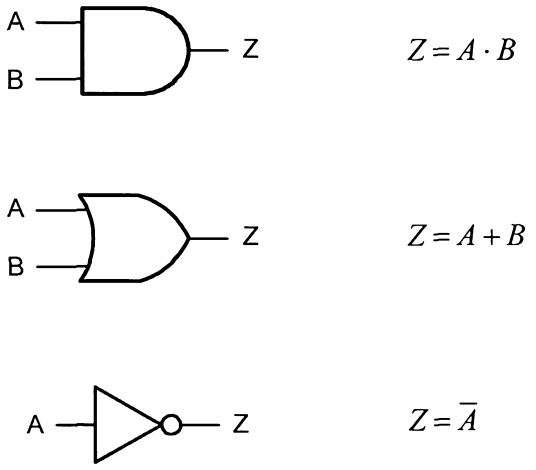


Figure 1 *The Boolean expressions for the AND, OR and NOT gates*

The Boolean expression for the AND gate is often abbreviated by leaving the dot out. So, $Z = A \cdot B$ becomes $Z = AB$.

It isn't necessary to have unique Boolean signs for the NAND or NOR gates because their expressions can be derived using the existing Boolean signs. To demonstrate this, consider the combinational logic circuit in Figure 2 below. It is an AND gate followed by a NOT gate. By now, you're probably experienced enough to work out that this circuit implements the NAND function.

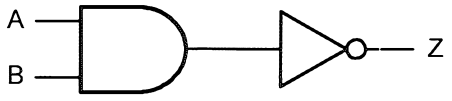


Figure 2 *A circuit that implements the NAND function*

Now let's analyse the circuit's operation using Boolean algebra. The output of the AND gate can be expressed as $Z = A \cdot B$ and the output of the NOT gate can be expressed as $Z = \overline{A'}$ (where A' is the input to the NOT gate). These expressions are shown on the circuit in Figure 3 below.

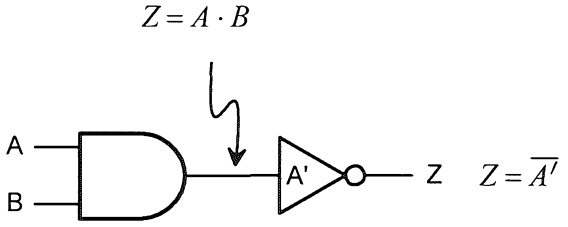


Figure 3 *The Boolean expressions that describe the outputs of the two gates*

As the input to the NOT gate is the same as the output of the AND gate (that is, $A' = A \cdot B$), the output of the circuit can be expressed as $Z = \overline{A \cdot B}$. Moreover, as the circuit implements the NAND function, this must also be the Boolean expression for the NAND gate.

So, the Boolean expressions for the NAND and NOR gates are as shown in Figure 4 below.

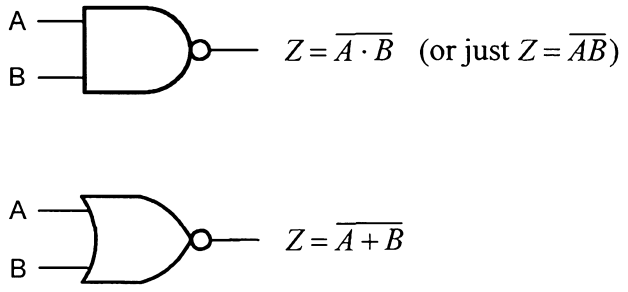


Figure 4 *The Boolean expressions for the NAND and NOR gates*

The XOR and XNOR functions do have their own special sign and these are shown in Figure 5 below. However, the sign is a relatively modern invention that cannot be manipulated using Boolean algebra and so they have limited use (as you will see).

Figure 5 *The Boolean expressions for the XOR and XNOR gates*

For gates with more than two inputs, simply add letters to the Boolean expression for each input separated by the sign for that gate (see Figure 6).

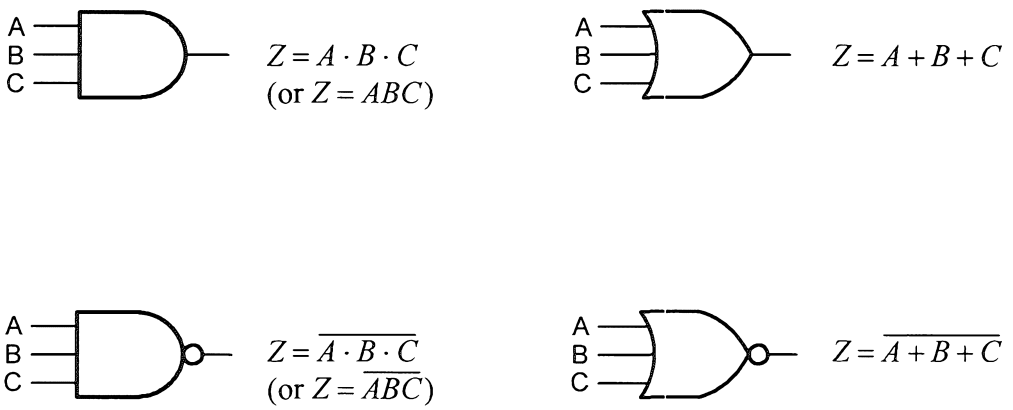


Figure 6 *The Boolean expressions for 3-input AND, OR, NAND and NOR gates*

Summary of the Boolean expressions

Function	OR	AND	NOR	NAND	XOR	XNOR
Symbol						
Expression	$Z = A + B$	$Z = A \cdot B$	$Z = \overline{A + B}$	$Z = \overline{A \cdot B}$	$Z = A \oplus B$	$Z = \overline{A \oplus B}$

Boolean expressions for combinational logic circuits

As you know, combinational logic circuits are made from individual logic gates connected together. So, combinational logic circuits can have Boolean expressions that describe their operation in the same way that individual logic gates do. An example of this is shown below in Figure 7.

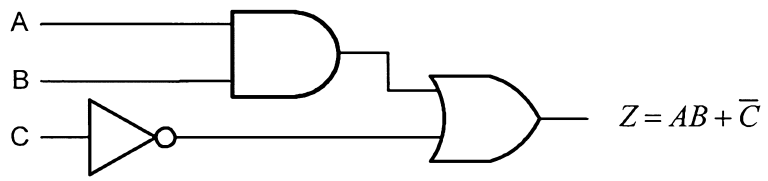


Figure 7 *A combinational logic circuit and the Boolean expression that describes its operation*

Writing Boolean expressions for combinational logic circuits

The process for writing a Boolean expression for a combinational logic circuit is surprisingly simple and is as follows:

1. Write the Boolean expression for the output of each gate in the first level of the circuit. This is demonstrated using the circuit in Figure 7 which is redrawn in Figure 8 below.

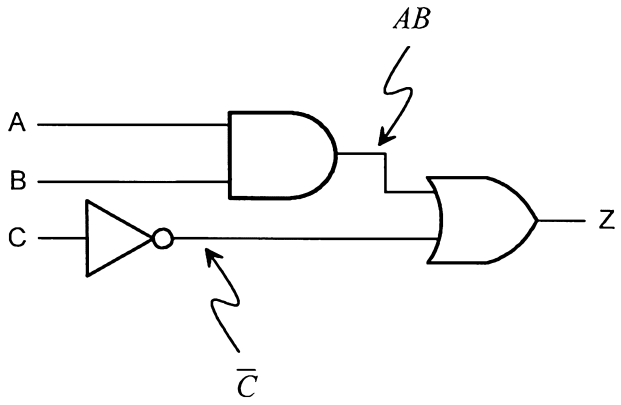


Figure 8

2. Write the Boolean expression for the output of each gate in subsequent levels substituting the letters for the gate's input or inputs with the Boolean expression for the node that the input is connected to. This is demonstrated in Figure 9 below.

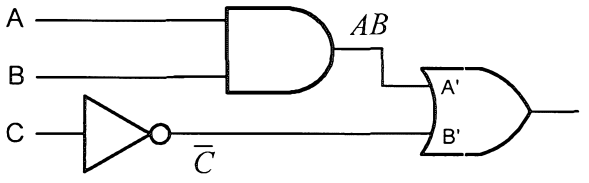


Figure 9

$Z = A' + B'$

But, as $A' = AB$...

and $B' = \overline{C}$ then...

$Z = AB + \overline{C}$

This procedure is demonstrated again using the three level combinational logic circuit in Figure 10 below.

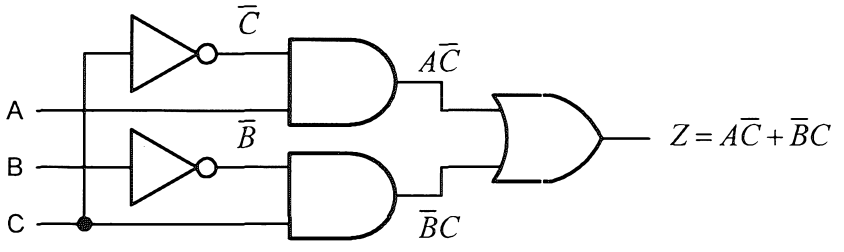


Figure 10 *Writing the Boolean expression of a three level combinational logic circuit*

Practise this procedure for yourself by writing the Boolean expression for *Z* for each of the following circuits.

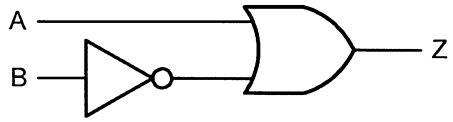


Figure 11

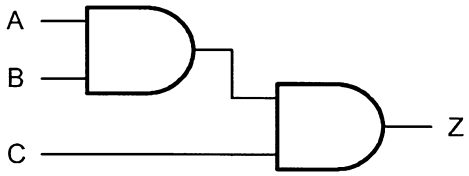


Figure 12

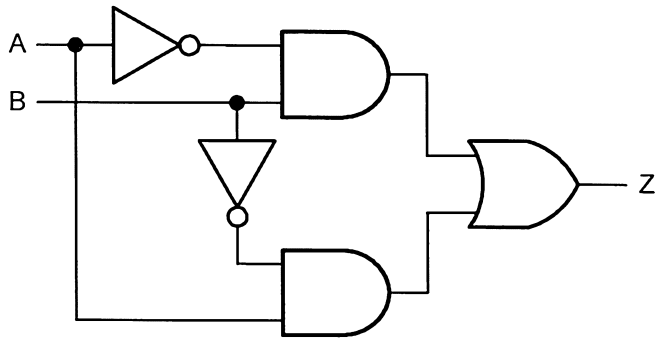


Figure 13

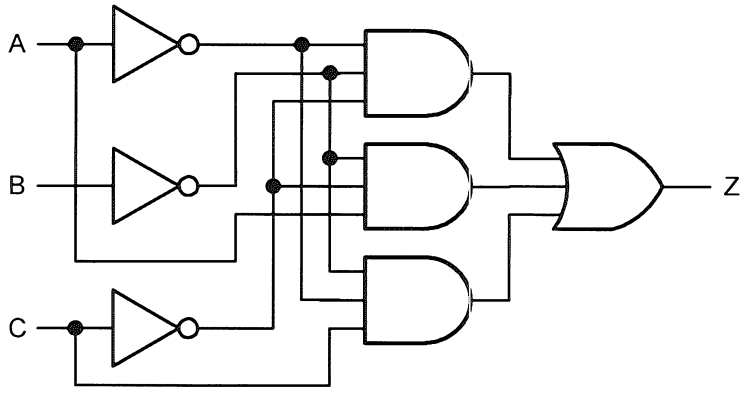


Figure 14

What's the point of describing combinational logic circuits using Boolean?

I'm glad you asked! Writing Boolean expressions for combinational logic circuits is much more than just an exercise in mathematics. It's very useful because, when the expression takes the correct form, it can be used to write the circuit's truth table directly without having to predict 1s and 0s through it for every input combination.

The form that the Boolean expression must take is a *sum-of-products* statement (or, more correctly, an *OR-of-ANDs* statement). An example is shown below:

$$Z = \overline{A}BC + A\overline{B}\overline{C}$$

This is called a sum-of-products expression because, if we think of conventional algebra, the expression looks like the addition of sets of variables that are being multiplied together.

To understand how to write the truth table for a sum-of-products expression it is useful to go back to basics and compare the Boolean expression for an AND gate with its truth table.

As we know, the Boolean expression for an AND gate is $Z=AB$. This expression tells us that the output is true when A and B are true. For all other input combinations the output is false. As you can see from the truth table for the AND gate (Table 1), the Z column only has a one in it where there are ones in both the A and B columns.

Table 1

B	A	Z
0	0	0
0	1	0
1	0	0
1	1	1

Using the same logic, if the expression is $Z = A\overline{B}$ then it tells us that the output is true when A is true and B is not true because the B has a NOT symbol written above it. This means that the Z column only has a one in it where A is a logic-1 and B is a logic-0. All other rows in the Z column are logic-0. This is shown in the truth table in Table 2.

Table 2

B	A	Z
0	0	0
0	1	1
1	0	0
1	1	0

Now, if we take a sum-of-products expression like $Z = AB + A\overline{B}$, it tells us that the output is true when A and B are true or when A is true and B is not true. For all other input combinations the output is false. So, the truth table for the expression looks like Table 3.

Table 3

B	A	Z	
0	0	0	
0	1	1	← The logic-1 in this row satisfies the $Z = A\overline{B}$ part of the expression.
1	0	0	
1	1	1	← The logic-1 in this row satisfies the $Z = AB$ part of the expression.

If we consider the example of a sum-of-products expression shown to you on page 4-10 (which is $Z = \overline{A}BC + A\overline{B}\overline{C}$) it can be demonstrated that the same principle applies to Boolean expressions with more than two inputs. The truth table for the expression is shown in Table 4.

Table 4

C	B	A	Z	
0	0	0	0	
0	0	1	1	← $Z = A\overline{B}\overline{C}$
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	← $Z = \overline{A}BC$
1	1	1	0	

Practise this process for yourself by writing the truth tables for Figures 11 to 14 from their sum-of-product Boolean expressions (rewritten below for you).

1. $Z = A + \overline{B}$

B	A	Z
0	0	
0	1	
1	0	
1	1	

2. $Z = \overline{A}B + A\overline{B}$

B	A	Z
0	0	
0	1	
1	0	
1	1	

3. $Z = ABC$

C	B	A	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

4. $Z = \overline{A}\overline{B}\overline{C} + A\overline{B}\overline{C} + \overline{A}\overline{B}C$

C	B	A	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Attempt the two questions below as well as they introduce you to slightly different examples of sum-of-product expressions.

5. $Z = AB\overline{C} + \overline{B}C$

C	B	A	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

6. $Z = ABC + \overline{C}$

C	B	A	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Simplifying Boolean expressions

Conveniently, the analysis of combinational logic circuits in Figures 11 to 14 produced sum-of-products Boolean expressions. However, you'll not always find this to be the case. For these situations, it's necessary to transform the expressions into the sum-of-products form (or a product-of-sums form but PoS expressions aren't discussed in this unit). There are a variety of techniques that can be used for this purpose and it would take several lessons to learn them all. For the purposes of this introductory unit we'll concentrate on Boolean simplification using DeMorgan's theorems.

DeMorgan's theorems

Augustus DeMorgan (an English mathematician born in 1807) was the first person to formalise two important theorems in symbolic logic (and so they were named after him).

DeMorgan's first theorem states that:

Two NOT signs across a variable or group of variables cancel

To demonstrate this theorem, let's consider the circuit shown in Figure 15 below with its expression for Z.

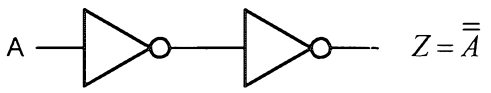


Figure 15

According to DeMorgan's theorem, the two NOT signs above the A cancel and so the expression simplifies to $Z = A$. This can be proven using the conventional method of circuit analysis introduced to you in the last section and shown below in Figure 16 and Table 5.

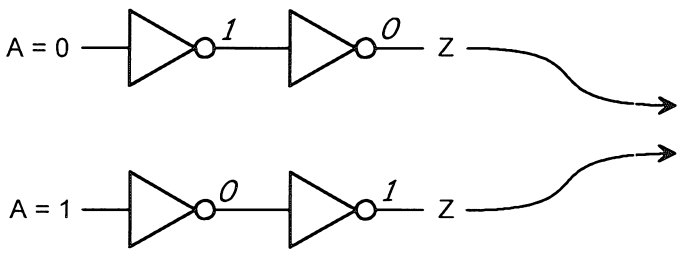


Table 5

A	Z
0	0
1	1

Figure 16 *The logic levels on the output of the circuit in Figure 15 for all combinations of inputs*

To demonstrate DeMorgan's first theorem again, let's consider the circuit in Figure 17 with its expression for Z.

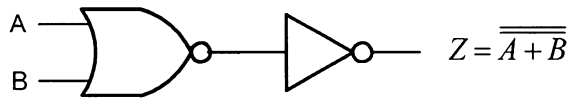


Figure 17

According to DeMorgan's theorem, the two NOT signs above $\overline{\overline{A + B}}$ cancel and so the expression simplifies to $Z = A + B$. Once the Boolean expression is simplified, we can see that the circuit implements the OR function and this can be proven if we analyse the circuit using conventional means as shown below in Figure 18 and Table 6.

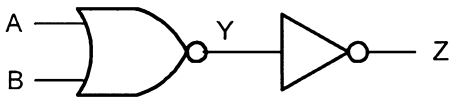


Figure 18

Table 6

B	A	Y	Z
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

Importantly, any cancelling of two NOT symbols can only be done when they cover the same variable or set of variables. The following examples are incorrect simplifications using this DeMorgan's theorem.

$Z = \overline{\overline{A}BC}$ does **not** simplify to $Z = ABC$

$Z = \overline{\overline{A}BC}$ does **not** simplify to $A\overline{BC}$

$Z = \overline{\overline{A}\,\overline{B}\,\overline{C}}$ does **not** simplify to $Z = ABC$

DeMorgan's second theorem simply states that:

If you break the line, change the sign (also, if you make the line, change the sign)

To demonstrate this theorem, let's consider the circuit shown in Figure 19 below with its expression for Z.

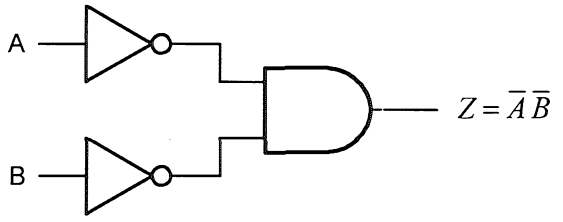


Figure 19

According to DeMorgan's theorem, the NOT signs above the letters can be made (that is, joined) after which the sign that it crosses must changed and so the expression simplifies to $Z = \overline{A + B}$. Once the Boolean expression is simplified, we can see that the circuit implements the NOR function and this can be proven using conventional means as shown below in Figure 20 and Table 7.

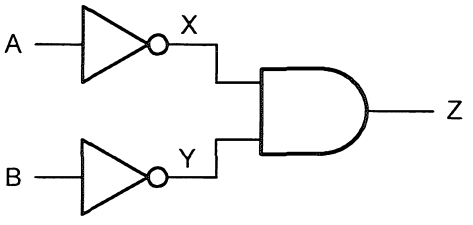


Figure 20

Table 7

B	A	X	Y	Z
0	0	1	1	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	0

Not surprisingly, there are Boolean expressions that need both of DeMorgan's theorems to be simplified. To demonstrate this, let's consider the circuit shown in Figure 21 below with its expression for Z.

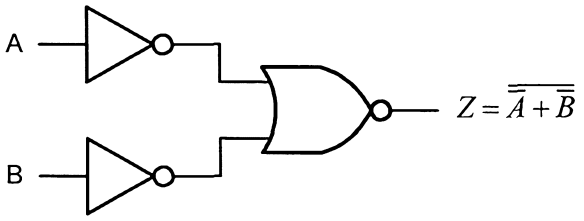


Figure 21

According to DeMorgan's theorems, the expression first simplifies to $Z = \overline{\overline{A}}\,\overline{\overline{B}}$ and then to $Z = AB$. So, we can see that the circuit implements the AND function.

Let's do one more example using the circuit shown in Figure 22 below with its expression for Z.

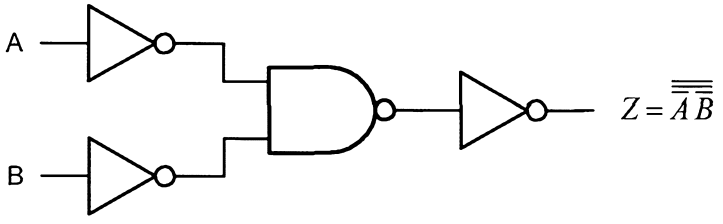


Figure 22

According to DeMorgan's theorems, the expression first simplifies to $Z = \overline{A}\,\overline{B}$ and then to $Z = \overline{A + B}$. So, we can see that the circuit implements the NOR function.

Practise using DeMorgan's theorems to simplify the following Boolean expressions.

1. $Z = \overline{\overline{A}BC}$

2. $Z = \overline{\overline{A}\,\overline{B}\,\overline{C}}$

3. $Z = \overline{A + \overline{B}\,\overline{C}}$

4. $Z = \overline{\overline{A + BC}}$

5. $Z = \overline{(A + \overline{B})(\overline{A} + C)}$

Practising analysing combinational logic circuits by developing the truth table for the following circuits by: 1) Writing the Boolean expression for the circuit; then 2) Where necessary, DeMorganising the expression to turn it into a sum-of-products statement; and 3) Writing the truth table from the SoP statement.

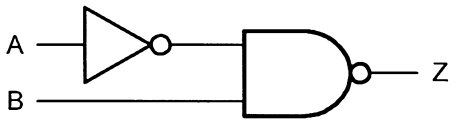


Figure 23

Working space

B	A	Z
0	0	
0	1	
1	0	
1	1	

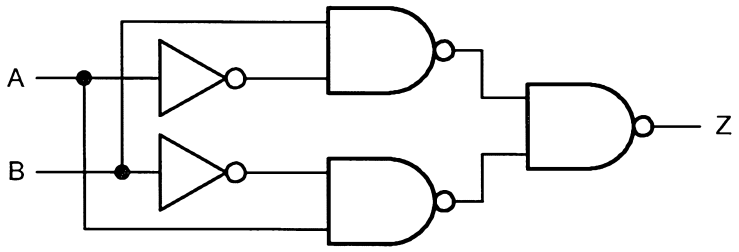


Figure 24

Working space

B	A	Z
0	0	
0	1	
1	0	
1	1	

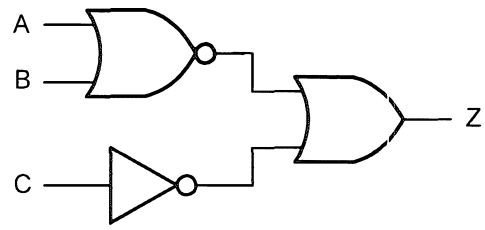


Figure 25

Working space

C	B	A	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

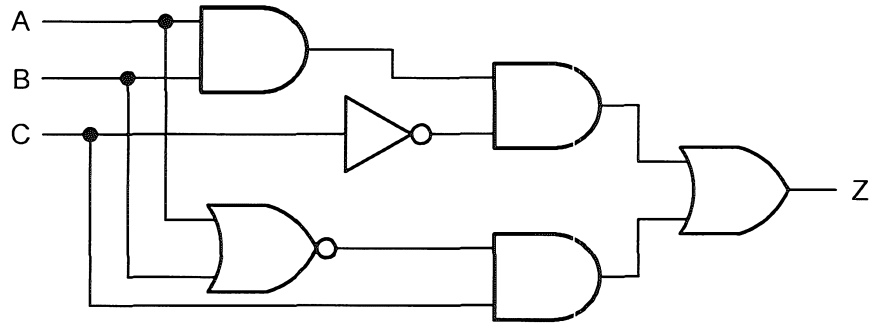


Figure 26

Working space

C	B	A	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Skill practice 4

Practise testing and measuring combinational logic digital circuits

This exercise is a practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 1½ hours to complete this exercise.

Equipment

- Digital trainer
- DMM
- logic probe
- 7400*
- 7402*
- 7404*
- 7408*
- 7432*
- hook-up wire (terminated)
- IC extractor

* The part number of the ICs that you will be using may have letters after the 74 (for example, 74LS00 or 74H00). These letters can be ignored for the purpose of this practical.

Remember:

Follow TAFE NSW WHS guidance at all times!

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____	Other: _____
_____	_____
_____	_____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

2. Gather the equipment needed for this exercise.

3. Set up +5V and ground rails on one of the digital trainer's breadboard like you did for the practical in Section 1 (see page 1-30).

4. For the circuit in Figure 1 (on the next page), write the Boolean expression for the output (Z) and, if necessary, simplify it to a sum-of-products expression.

5. Use the Boolean expression to predict the theoretical Z for all input conditions. Record this in the appropriate column of Table 1 (also on the next page).

6. Wire the circuit on the digital trainer making sure that you:

   - Write the IC pin numbers of the gates you work with on the circuit that you're instructed to wire. Remember, pinout data for these ICs is at the back of the book.
   - Handle all ICs carefully - remember ESD can damage them.
   - Wire each of the circuit's inputs to a toggle switch.
   - Ensure the power supply is connected to all ICs in the circuit.
   - When plugging hook-up wire into the breadboard to make connections to the pins of the IC, use holes that are furthest away from the IC's pin.

7. Determine the circuit's truth table experimentally recording your results in the appropriate column of Table 1.

8. Compare your predicted and measured values for Z. Do they agree with one another?

   Note 1: If they don't, you'll need to check either your theoretically predicted Z or your circuit wiring (or both!).

   Note 2: If they do, call the teacher to demonstrate the working circuit.

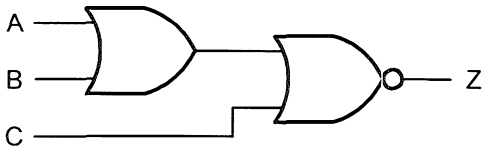
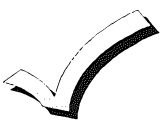


Figure 1

Z = _____

Table 1

Input C	Input B	Input A	Theoretical output (Z)	Measured output (Z)
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



The teacher needs to check your work at this point...

9. Repeat this procedure for the following circuits.

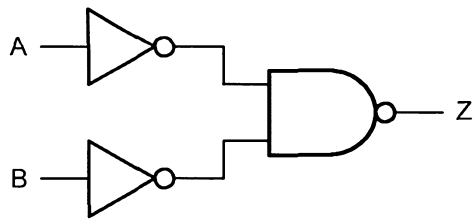
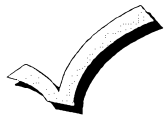


Figure 2

Z = _____

Table 2

Input B	Input A	Theoretical output (Z)	Measured output (Z)
0	0		
0	1		
1	0		
1	1		



The teacher needs to check your work at this point...

Copyright: Not to be reproduced without permission

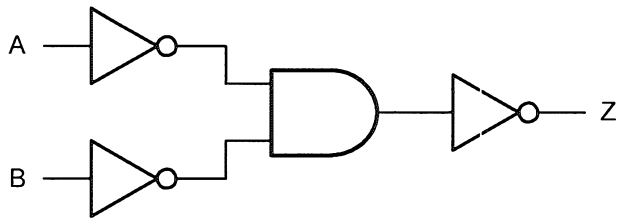
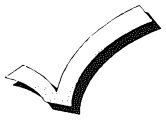


Figure 3

Z = _____

Table 3

Input B	Input A	Theoretical output (Z)	Measured output (Z)
0	0		
0	1		
1	0		
1	1		



The teacher needs to check your work at this point...

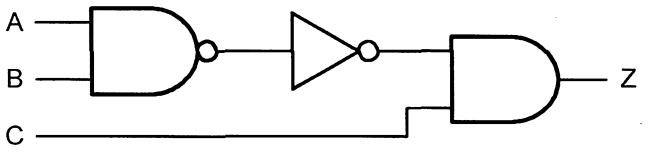
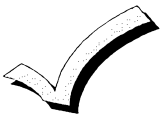


Figure 4

Z = _____

Table 4

Input C	Input B	Input A	Theoretical output (Z)	Measured output (Z)
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



The teacher needs to check your work at this point...

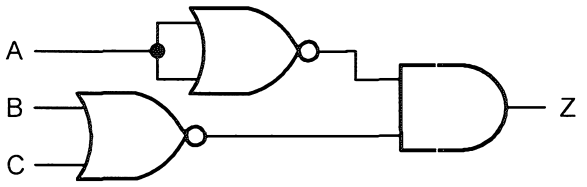
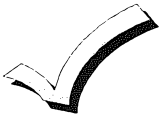


Figure 5

Z = _____

Table 5

Input C	Input B	Input A	Theoretical output (Z)	Measured output (Z)
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



The teacher needs to check your work at this point...

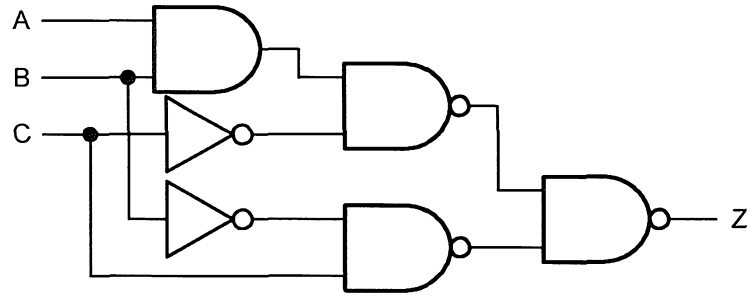
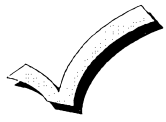


Figure 6

Z = _____

Table 6

Input C	Input B	Input A	Theoretical output (Z)	Measured output (Z)
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



The teacher needs to check your work at this point...

Copyright: Not to be reproduced without permission

Student notes

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. Write the Boolean expressions for the following gates:

   (a) AND _____

   (b) NOT _____

2. What gates do the following Boolean expressions represent?

   (a) $\overline{AB}$ _____

   (b) $A+B$ _____

   (c) $A \oplus B$ _____

3. Apply DeMorgan's theorems to the following Boolean expressions. (The number in brackets indicates how many times you're to use DeMorgan's theorem for each.)

   (a) $Z=\overline{\overline{AB}}$ (1)

   (b) $Z=\overline{A}+\overline{B}$ (1)

   (c) $Z=\overline{\overline{A}+B}$ (2)

(d) $Z = \overline{A\overline{B}}$ (2)

(e) $Z = \overline{A}\,\overline{\overline{B}}$ (1)

(f) $Z = \overline{\overline{A}\overline{B}C}$ (2)

(g) $Z = \overline{AB} + \overline{C}$ (1)

(h) $Z = \overline{A} + \overline{B + C}$ (1)

(i) $Z = \overline{\overline{A} + \overline{B}C}$ (2)

(j) $Z = \overline{\overline{A}B\overline{C}}$ (2)

4. Write the Boolean expressions for the following circuits (but don't DeMorganise them):

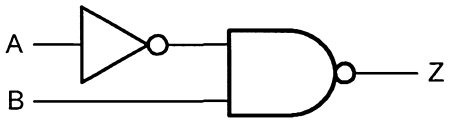


Figure 1

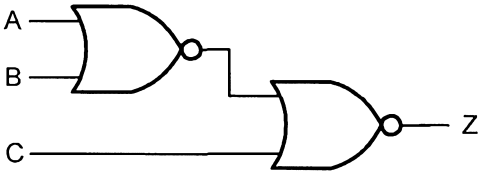


Figure 2

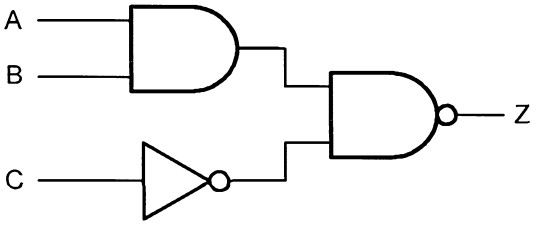


Figure 3

Copyright: Not to be reproduced without permission

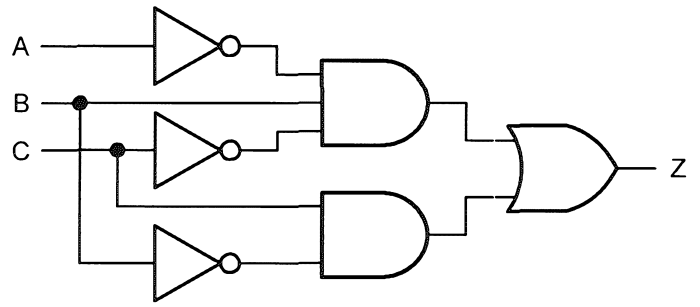


Figure 4

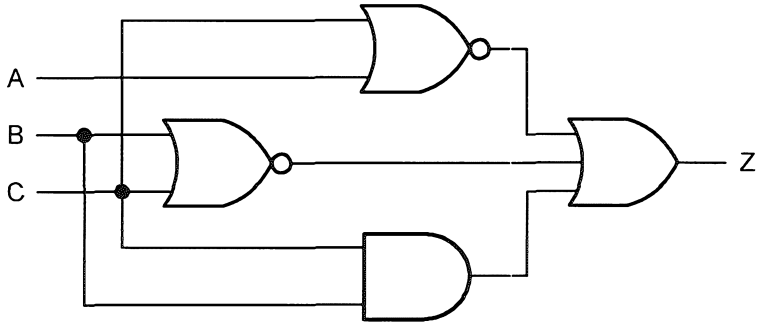


Figure 5

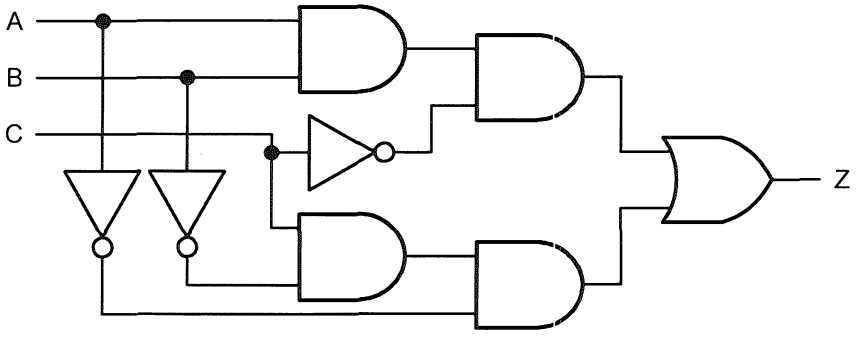


Figure 6

5. Draw the circuit that implements the Boolean expression $Z = \overline{A} + \overline{B} + \overline{C}$.

6. Draw the circuit that implements the Boolean expression $Z = A\overline{C} + \overline{B}C$.

7. Complete the truth table for the following sum-of-products Boolean expressions.

a) $Z = A\overline{B} + AB$

B	A	Z
0	0	
0	1	
1	0	
1	1	

b) $Z = \overline{A}\overline{B} + \overline{A}B + A\overline{B}$

B	A	Z
0	0	
0	1	
1	0	
1	1	

c) $Z = A\overline{B}\overline{C} + \overline{A}\overline{B}C + A\overline{B}C + \overline{A}BC$

C	B	A	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

d) $Z = \overline{A}\overline{B} + B\overline{C}$

C	B	A	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

8. Develop the truth table for the combinational logic circuit in Figure 7 using Boolean algebra.

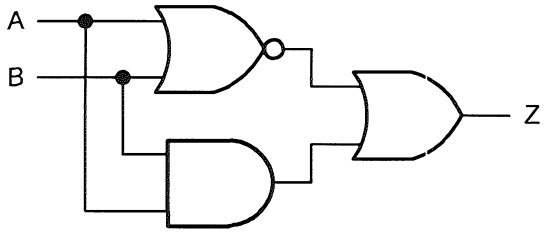


Figure 7

Working space

B	A	Z
0	0	
0	1	
1	0	
1	1	

9. Develop the truth table for the combinational logic circuit in Figure 8 using Boolean algebra.

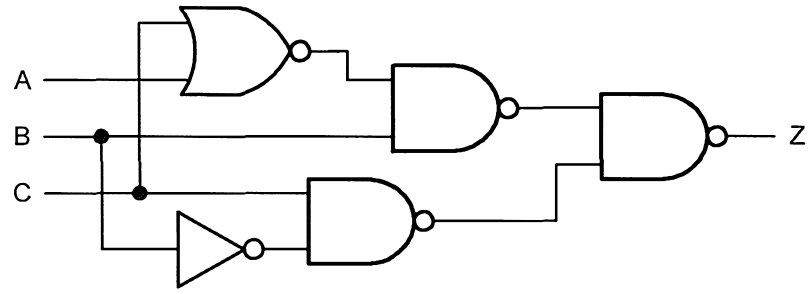


Figure 8

Working space

C	B	A	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Copyright: Not to be reproduced without permission

Student notes

Section 5 Logic families

Purpose To deepen your understanding of the operation of integrated circuit logic devices to develop your ability to interpret the manufacturers' data sheets of logic gate ICs.

Objectives At the end of this section you should be able to:

- Identify the logic family and sub-family of a digital logic IC given its part number
- State the power supply requirements for commercial grade TTL and CMOS families and sub-families
- Define the following digital logic device performance characteristics: *propagation delay*, *power dissipation*, *maximum clock speed*, *fanout*, *unit load* and *noise margin*
- Identify these performance characteristics for commercial grade TTL and CMOS families and sub-families given data
- Identify the advantages and disadvantages of TTL sub-families when compared with each other given data
- State the advantages and disadvantages of TTL and CMOS logic devices when compared with each other given data
- Identify the input and output voltage and current specifications for TTL and CMOS sub-families given data
- Recall the input and output voltage specifications for the 74ALS series TTL and 4000 series CMOS
- Define the terms *sinking* and *sourcing* with regard to logic device output terminal currents
- Choose an appropriate substitute for a digital logic IC given data
- Recognise that TTL and 4000 series CMOS are not readily compatible
- Identify interfacing strategies required to let TTL logic devices directly or indirectly drive 4000 series CMOS devices and vice versa
- Use a DMM and logic probe to test and measure mixed family combinational logic circuits

Introduction

Logic gates (and the other logic devices that you will learn about in this subject) are implemented by extremely small electronic circuits inside the IC. Importantly, there is always more than one way to implement the function of a logic gate. Moreover, different circuit designs rely on different types of components. For example, while the function of a NAND gate can be implemented by a circuit using just resistors and diodes, it can also be implemented by a circuit using resistors and a single transistor or another circuit using several transistors and still another circuit using several FETs. Regardless of the components used, if the circuit is a NAND gate then it will have the truth table of a NAND gate.

However, implementing logic gates using different technologies usually varies some of the gate's important performance and electrical specifications. It is important for technicians to know about and understand these specifications and how they differ. We'll look at these issues in this section.

Logic families

A *logic family* is a group of digital ICs that are all based on the same (or very similar) technology. For example, an early technology of implementing logic gates from circuits made resistors and diodes are said to belong to a family of logic devices called *Resistor Diode Logic* (*RDL or DRL*). Logic gates made from circuits using resistors and a transistor are said to belong to a family of logic devices called *Resistor Transistor Logic* (*RTL*).

There are several logic families but in this workbook we'll concentrate on just two of the more enduring and readily available logic families:

- *Transistor Transistor Logic* (*TTL*)
- *Complementary Metal Oxide Semiconductor Logic* (*CMOS*)

This section looks at the similarity and differences between TTL and CMOS and considers the advantages and disadvantages of each. It also looks at the problems associated with interconnecting between TTL and CMOS logic circuits.

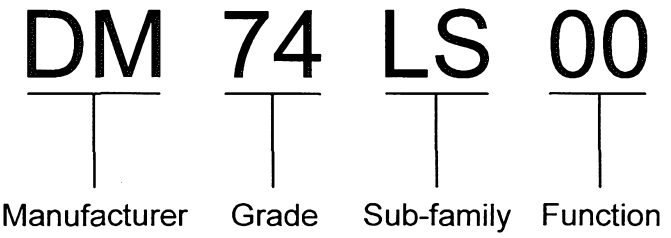
Transistor Transistor Logic (TTL)

The TTL part number system

The part number of TTL devices usually takes the form of the example shown directly below.

DM74LS00

The part number communicates a great deal about the IC and the breakdown is shown below.



The letters at the front of the part number denote the manufacturer of the IC. For example, the letters DM tells us that the IC is made by National Semiconductors. Texas Instruments uses SN and Motorola uses MC.

The next two digits denote the grade of the TTL IC. "74" is used to denote that the IC is commercial grade and "54" denotes that the IC is military grade. Military grade ICs are more robust (that is, they're tougher) than commercial grade chips. In particular, they can operate over a wider range of temperatures (-55°C to +125°C instead of 0°C to +70°C). Naturally, manufacturers of equipment for the armed forces might prefer to use military these grade components because the operating conditions of their equipment can be very severe and reliability essential.

The next letters denote the IC's sub-family. Sub-families will be discussed later in this section.

The last two or three digits denote the IC's function. For example, "00" denotes that a TTL IC is a *quad two-input NAND gate*. That is, the IC contains four NAND gates, each with two inputs. If the last two digits were "02" then the IC would be a *quad two-input NOR gate* and an "04" is a hex NOT gates.

Technicians are expected to be able to work out whether an IC is commercial grade or military grade as well as identifying the IC's sub-family without referring to data. However, technicians would normally need data to determine the manufacturer of an IC and/or its function.

TTL power supply requirements

As you have probably already learnt, all logic ICs need power to operate. There are usually two pins on every IC dedicated to this purpose and on TTL ICs these pins are normally referred to as V_{CC} and ground. This is shown in Figure 1 below where pin-14 is V_{CC} and pin-7 is ground. It is important to point out here that the two pins at these ends of the IC are not always the power supply pins.

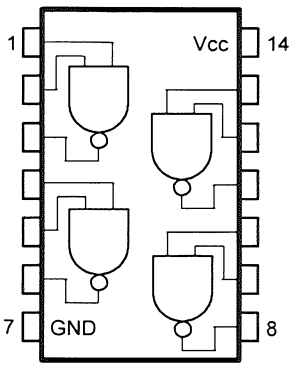


Figure 1 *The Vcc and ground pins on a typical TTL IC*

Inside the IC the power supply connects to every gate to provide them with power. These internal connections are not usually shown in pinout data because it is understood that they're always there.

The ground pin on TTL ICs is connected to the circuit's common rail (which may or may not actually be connected to ground) and the V_{CC} pin is connected to the positive voltage supply rail. TTL ICs usually require +5V for the V_{CC} but there is slight difference between commercial grade (74 series) and military grade (54 series) TTL ICs in this regard. This difference is shown in Table 1 below.

Table 1

Grade	*Vcc* voltage	Maximum allowable variation in *Vcc*	Range of allowable *Vcc* voltages
74 Series	5V	±5%	4.75V to 5.25V
54 Series	5V	±10%	4.5V to 5.5V

From the data in Table 1 it can bee seen that the military grade version of TTL operates over a wider range of supply voltages making their operation more reliable under larger power supply voltage variations.

TTL sub-families

With modifications to the standard TTL IC's internal circuit design, several variations of TTL are possible. These variations are known as *sub-families*. The sub-family type is denoted by the letters that follow the 74 or 54. TTL sub-families that have been commercially available include:

- **74** The standard or original TTL IC
- **74L** A low power version (but slower)
- **74H** A high speed version (but higher power usage)
- **74S** A version using Schottky technology
- **74LS** A low power version using Schottky technology
- **74AS** A version using advanced Schottky technology
- **74ALS** A low power version using advanced Schottky technology

Table 2 below shows how these sub-families vary in several key performance characteristics.

Table 2	Sub-family						
Performance characteristic	**74**	**74L**	**74H**	**74S**	**74LS**	**74AS**	**74ALS**
Propagation delay (ns)	9	33	6	3	9.5	1.7	4
Power dissipation (mW/gate)	10	1	23	20	2	8	1.2
Maximum clock rate (MHz)	35	3	50	125	45	200	70
Fanout	10	20	10	20	20	40	20

Let's examine these performance characteristics.

Propagation delay

The *propagation delay* (t_{pd}) of a gate is what we would call for humans their "reaction time". For example, the propagation delay of the standard 74 sub-family is typically 9 nanoseconds which means that the output changes 9 nanoseconds after the input changes (assuming the new logic level on its inputs will result in a change on the output in the first instance). Figure 2 below shows how the sub-families compare in this regard.

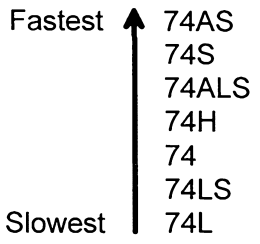


Figure 2 *The TTL sub-families ranked from slowest to fastest*

Power dissipation

The quoted power dissipation of a sub-family is a rating per IC or per gate depending on the IC. The figures quote in Table 2 are per gate. So, the 74H00 for example is a high speed version of a quad two input NAND gate and so it contains four NAND gates that dissipate 23mW/gate. This means that the total power dissipated by the IC is 92mW.

This power rating of TTL ICs is largely independent of the speed at which they're operated. That is, each gate will dissipate its rated power regardless of whether the device's output is changing twice a day or one million times a second (1MHz).

Figure 3 below shows the sub-families compare.

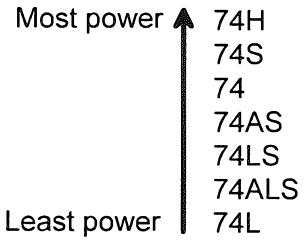


Figure 3 *The TTL sub-families ranked from lowest to highest power consumption*

Maximum clock speed

The maximum clock speed of a logic gate is the highest rated frequency pulse train that can be connected to a gate's input and still allow the gate to produce a pulse train on its output that would be recognisable by the input of another gate in the same sub-family. For example, Figure 4 below shows what would happen when a 5MHz pulse train is connected to the input of a NOT gate in a 7404 which has a maximum clock speed of 35MHz and a NOT gate in a 74L04 which has a maximum clock speed of only 3MHz.

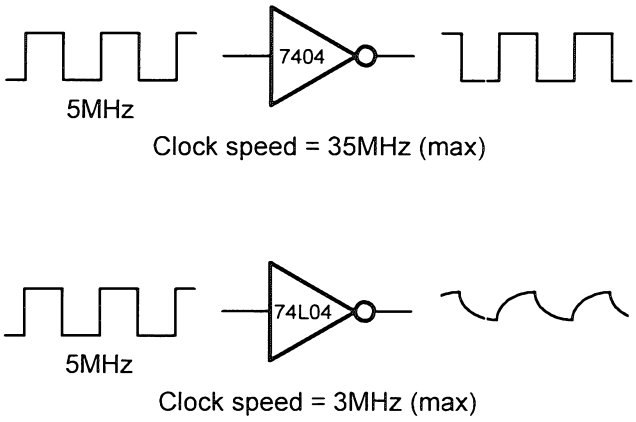


Figure 4 *The outputs of a 7404 and a 74L04 with a 5MHz pulse train connected to their inputs*

Maximum clock speed is an important performance characteristic of digital ICs because some digital circuits use digital signals with very high frequencies (for example, personal computers) so the logic gate in these circuits must be able to operate at these frequencies.

Fanout

Fanout is the name for the number of logic device inputs that a logic device output can drive. For example, the standard 74 series has a fanout of ten. This means that one standard 74 series TTL logic gate's output can drive ten other standard 74 series logic gate inputs. An example of this is shown in Figure 5 below using NOR gates. Notice that the NOR gate is driving ten inputs and not ten gates!

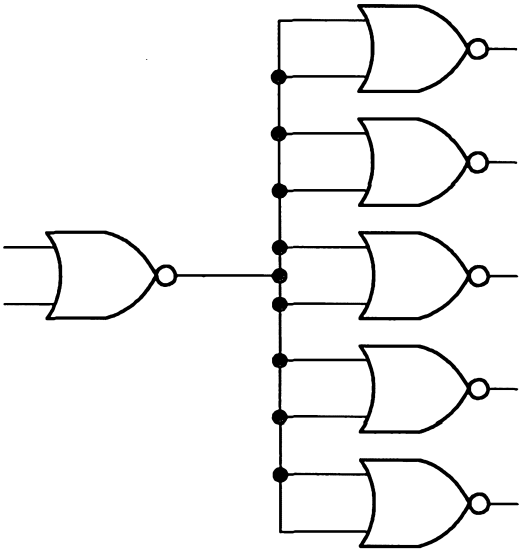


Figure 5 *A 74 standard series TTL gate has a fanout of 10 which means that its output can drive ten other 74 standard series TTL inputs*

Importantly, the specifications for fanout given in Table 2 are only valid if the gate is driving inputs to logic devices of the same sub-family. This means that the fanout for a standard 74 series TTL logic gate will not be ten if it is driving inputs to logic gates from other sub-families.

It is possible to calculate the new fanout of a TTL sub-family when driving a different sub-family but you'll not be asked to do so for the assessments for this unit.

Unit load

Logic device inputs are often technically referred to as *unit loads*. So, as the standard 74 series has a fanout of ten, we would say that it's capable of driving up to ten unit loads at once. The number of unit loads an output is capable of driving is a function of the output's sink and source current capability and the input's current needs. More on this on the next few pages.

TTL logic level voltage specifications

Previously, you've been taught that a logic-0 is 0V and a logic-1 is 5V but this isn't strictly true. These are ideal TTL voltages and in practice a wide range of voltages are acceptable for both logic levels. Table 3 below shows the input and output voltage specifications for the readily available TTL sub-families for logic-0 and logic-1. You're not expected to memorise all of this data but you are expected to know the specifications for the 74ALS sub-family.

Table 3	Sub-family			
Voltage specification (in V)	**74**	**74LS**	**74AS**	**74ALS**
Logic-1 output voltage - V_{OH} (min)	2.4	2.7	2.5	2.5
Logic-0 output voltage - V_{OL} (max)	0.4	0.5	0.5	0.4
Logic-1 input voltage - V_{IH} (min)	2.0	2.0	2.0	2.0
Logic-0 input voltage - V_{IL} (max)	0.8	0.8	0.8	0.8

To help explain this data, let's consider the 74ALS series as an example.

Although the ideal TTL voltage for a logic-1 is 5V, this data tells us that, in practice, when the output of a 74ALS series logic device is a logic-1 the output voltage (V_{OH}) can be as low as 2.5V. In other words, if the gate's output is any voltage equal to or greater than 2.5V then it is considered a valid logic-1. Similarly, although the ideal TTL voltage for a logic-0 is 0V, in practice, when the output of a 74ALS series is a logic-0 the output voltage (V_{OL}) can be as high as 0.4V. This is illustrated in the diagram in Figure 6 below.

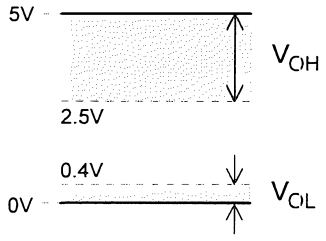


Figure 6 *The range of acceptable voltages for V_{OH} and V_{OL} for 74ALS series TTL*

The inputs to logic gates operate in a similar way. For example, the data for the 74ALS series tells us that it will accept any voltage on the input above 2V as a logic-1 (V_{IH}) and any voltage below 0.8V as a logic-0 (V_{IL}). This is illustrated in the diagram in Figure 7 below.

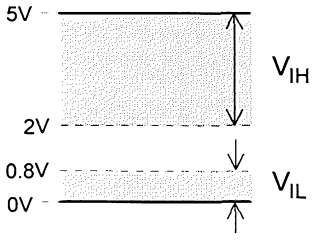


Figure 7 *The range of acceptable voltages for V_{IH} and V_{IL} for 74ALS series TTL*

Notice in the diagram in Figure 7 that there is a range of input voltages that are not considered either a logic-0 or a logic-1. This region is known as the *transition zone* and is illustrated in Figure 8 below.

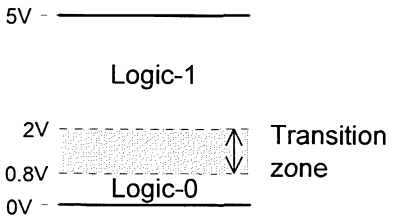


Figure 8 *The transition zone for 74ALS series TTL*

All logic devices - except those with *Schmitt trigger* inputs - have a transition zone and, as we can see from Figure 8, the transition zone for the 74ALS series is any voltage greater than 0.8V but lower than 2V. If a voltage is applied to an input between these voltages the logic level on the output is considered to be *indeterminate*. That is, the manufacturers don't guarantee what logic level the input voltage will be read as.

Notice also that there is a difference between the acceptable input and output voltages for the two logic levels. Specifically, the inputs are more forgiving and will accept lower than minimum output voltages for logic-1s and higher than maximum output voltages for logic-0s. This difference is called *noise margin* and is provided so that if the digital signal picks up some low level electrical noise the voltages on the logic device's inputs aren't forced into its transition zone.

TTL logic level current specifications

As just discussed, the logic state on an input or output of a logic device is defined by its voltage. However, recall that logic gates and other logic devices are implemented by electronic circuits. This means that, when a logic device of any sort is powered up and connected in a logic circuit, current must flow around the circuit.

When a logic device's output is a logic-1, conventional current flows in the direction shown in Figure 9 below. As you can see, current flows out of the driving device's output and into the driven device's input and this is true regardless of the logic gate or device. Under these conditions, the driving device is said to be *sourcing* current.

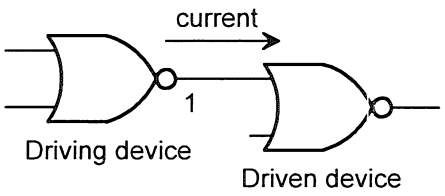


Figure 9 *Current flows in the direction shown when a logic circuit's node is logic-1*

When a logic device's output is a logic-0, conventional current flows in the direction shown in Figure 10 below. As you can see, current flows out of the driven device's input and into the driving device's output. Under these conditions, the driving device is said to be *sinking* current.

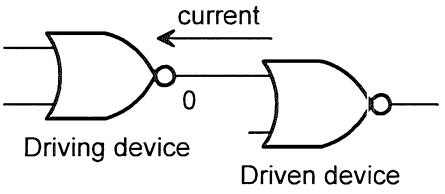


Figure 10 *Current flows in the direction shown when a logic circuit's node is logic-0*

The manufacturers of logic devices usually specify the maximum amount of current that flows in or out of a logic device's input as well as the maximum amount of current that its output can source or sink. The current specifications for readily available TTL sub-families are summarised in Table 4 below. You're not expected to memorise this data.

Table 4

	Sub-family			
Current specification (in mA)	**74**	**74LS**	**74AS**	**74ALS**
Logic-1 output current - I_{OH} (max)	0.4	0.4	1.0	0.4
Logic-0 output current - I_{OL} (max)	-16	-8	-20	-8
Logic-1 input current - I_{IH} (max)	40µA	20µA	50µA	20µA
Logic-0 input current - I_{IL} (max)	-1.6	-0.4	-2.0	-0.4

To help explain the data in this table, let's consider the 74ALS series as an example. The maximum current that can be sourced when the output of a 74ALS logic device is a logic-1 (I_{OH}) is 0.4A or 400µA. The maximum input current needed when the input of a 74ALS logic device is a logic-1 (I_{OH}) is 20µA. These currents are shown in Figure 11 below.

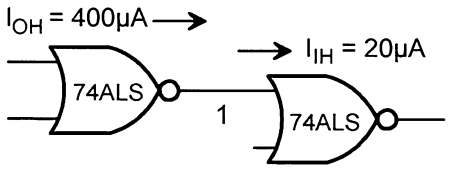


Figure 11 *The I_{OH} and I_{IH} for the 74ALS series TTL*

From this information, you can see why the fanout of the 74ALS series TTL is twenty (ie 400µA ÷ 20µA = 20).

As mentioned a few pages back, logic device inputs are referred to as "unit loads" and are parametrically expressed in terms of the device's input current requirements. So we can say that the 74ALS family's logic-1 *unit load* is 20µA, in which case the 74ALS is capable of driving twenty 74ALS logic-1 unit loads.

The data also tells us that the maximum current that can be sunk when the output of a 74ALS logic device is a logic-0 (I_{OL}) is 8mA. The maximum input current needed when the input of a 74ALS logic device is a logic-0 (I_{IL}) is 0.4mA. These currents are shown in Figure 12 below.

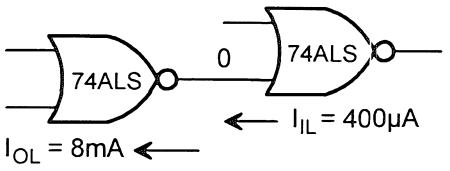


Figure 12 *The I_{OL} and I_{IL} for the 74ALS series TTL*

Again, from this information, you can see why the fanout of the 74ALS series TTL is twenty (ie 8mA ÷ 4000µA = 20). And we can say that the 74ALS family's logic-0 *unit load* is 400µA, in which case the 74ALS is capable of driving twenty 74ALS logic-0 unit loads.

Replacing faulty TTL ICs

When replacing faulty components in electronics equipment, it is common for technicians to use substitute devices instead of original components. This occurs for several reasons including:

- the original component is no-longer available
- the original component is over-priced
- the repair is urgent and the original component is not immediately available

Using substitute components (for the above mentioned reasons) is an acceptable practice in electronics as long as the substitute has specifications that are equal to or better than the original component (or the specifications are sufficient for the circuit conditions but a decision in this regard requires accurate knowledge of the circuit conditions in all circumstances).

At first glance, choosing an appropriate substitute for a faulty TTL logic device is easy. As the TTL part-numbering system is standardised across the TTL sub-families, any sub-family can potentially be substituted for another. For example, a 7400, 74LS00, 74AS and a 74ALS00 all do the same job. That being the case, it is not unreasonable to think that these ICs can substitute for one another.

However, the issue is a little more problematic than that and so choosing appropriate substitute components can require a lot of electronics knowledge and experience. As already discussed, there are slight variations in the performance and electrical specifications between the TTL sub-families which means that some sub-families cannot always substitute for others. For example, in some situations a 74LS00 cannot substitute for a 7400 but a 74AS00 can. Replacing a TTL logic device with an inappropriate sub-family can result in the circuit still not working or, perhaps worse, no working intermittently.

The following is a guide that you can use to help you decide whether a TTL logic device from one sub-family will substitute for another.

- Propagation delay - the substitute sub-family should have a propagation delay equal to, or shorter than, the original part.

- Maximum clock frequency - the substitute sub-family should have a maximum clock frequency equal to, or higher than, the original part.

- Input-output current specifications - there are two aspects to these specifications to consider:

  The substitute sub-family should have I_{OH} and I_{OL} specifications that are equal to, or higher than, the original part. Tip: In all but one case (the 74H series) the faster the sub-family, the higher I_{OH} and I_{OL} the specifications.

  The substitute sub-family should have I_{IH} and I_{IL} specifications that are equal to, or lower than, the original part.

- Power dissipation - if the equipment is battery operated, the substitute sub-family should have a power dissipation equal to, or less than, the original part. For mains operated equipment, the power dissipation is unlikely to be an important issue.

- Noise margin - this issue need only be checked if the electronics equipment usually operates in electrically noisy situations.

Importantly, the substitute sub-family should satisfy the first three criteria otherwise it may not be a suitable substitute. Having said that, the substitute may not need to satisfy these criteria if the original part had specifications more than what is actually needed for the circuit. Being able to determine if this is the case for a particular circuit requires a lot of experience.

Practise using these criteria to choose suitable substitutes for yourself by trying the following questions. To answer these questions you'll need to consider the data in Table 2 (page 5-5).

1. Explain why a 74ALS02 **is** a suitable substitute for a 74H02.

2. Explain why a 74S32 **may not** be a suitable substitute for a 74ALS32.

3. Explain why a 74L86 **may not** be a suitable substitute for a 74LS86.

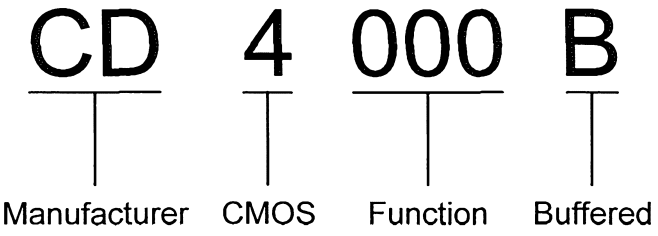
Complementary Metal Oxide Semiconductor Logic (CMOS)

The CMOS part number system

The part number of CMOS devices usually takes one of two forms. An example of the first form is shown directly below.

CD4000B

Broken down, the part number reads as shown below.



The letters at the front of the part number denote the manufacturer of the IC. The next digit denotes that the device is 4000 series CMOS (there are no military grade versions of 4000 series CMOS devices). The next three digits denote the IC's function. For example, "000" denotes that the CMOS IC is a *dual three-input NOR gate plus an inverter*. The last digit indicates that the CMOS chip is buffered to help protect the device from damage due to ESD. Most, if not all, CMOS devices are buffered these days so they'll usually have the letter B on the end.

Importantly, the function numbers of 4000 series CMOS are not the same as for TTL. That is, a 4000 does not do the same thing as a 7400, a 4032 does not do the same as a 7432 and so on.

The break-down of the second CMOS part number system will be dealt with when we discuss CMOS sub-families later this section.

4000 series CMOS power supply requirements

Like TTL logic devices, there are two pins on every CMOS IC dedicated to connecting power to the circuits inside. For 4000 series CMOS devices, these pins are normally referred to as V_{DD} and ground.

The ground pin on CMOS ICs is connected to the circuit's ground or common rail and the V_{DD} pin is connected to the positive voltage supply rail. Importantly, the 4000 series of CMOS ICs can operate over a wide range of supply voltages. Table 5 below shows the minimum and maximum supply voltages for 4000 series CMOS. The minimum and maximum supply voltages for commercial grade TTL are also shown for comparison.

Table 5

Logic family	Minimum	Maximum
4000 series CMOS	3V	18V
74 series TTL	4.75V	5.25V

This ability of 4000 series CMOS to operate over wider range of supply voltages makes it ideal for battery operated digital electronics equipment.

4000 series CMOS performance specifications

Table 6 below shows the performance specifications for 4000 series CMOS. The performance specifications of 74ALS series TTL are also shown for comparison.

Table 6

Performance characteristic	4000 series	74ALS series
Propagation delay (ns)	70	4
Power dissipation (mW)	0.01	1.2
Maximum clock rate (MHz)	5	70
Fanout	50	20

The quoted power dissipation figure in Table 6 is a static value. That is, each gate will consume 0.01mW (or 10µW) as long as the input and output logic levels are not changing. However, once these logic levels start changing, the power dissipation goes up (unlike TTL). Moreover, the faster they change, the greater the power dissipation. At 1MHz the power dissipation is typically about 1mW per gate.

If you compare the specifications of the 4000 series CMOS devices with 74ALS series TTL you'll notice that CMOS is not as fast as TTL. However, CMOS consumes much less power which is another reason why these devices are ideal for battery operated digital electronics equipment.

4000 series CMOS voltage specifications

Table 7 below shows the input and output voltage specifications for 4000 series CMOS. You are expected to recall these specifications. The voltage specifications of 74ALS series TTL are also shown for comparison.

Table 7

Voltage specifications	4000 series	74ALS series
Logic-1 output voltage - V_{OH} (min)	V_{DD}	2.5
Logic-0 output voltage - V_{OL} (max)	0V	0.4
Logic-1 input voltage - V_{IH} (min)	$0.7 \times V_{DD}$	2.0
Logic-0 input voltage - V_{IL} (max)	$0.3 \times V_{DD}$	0.8

There are two interesting things to note about the CMOS specifications in Table 7. First, the output voltages for logic-1 and logic-0 for 4000 series CMOS are ideal. That is, the output voltage for a logic-0 (V_{OL}) is 0V and the output voltage for a logic-1 (V_{OH}) is equal to the supply voltage (V_{DD}).

Second, with the exception of V_{OL}, specific values are not given for the logic-1 and logic-0 voltages. This is because the supply voltage may not necessarily be 5V. Instead, the voltages are specified in relation to the supply voltage. The V_{OH} is the same as V_{DD}, V_{IH} is equal to or greater than 70% of V_{DD} and V_{IL} is equal to or less than 30% of V_{DD}.

This means that, if a 4000 series CMOS logic device is operating on a supply voltage of 5V:

- the logic-1 output voltage (V_{OH}) is 5V
- the logic-0 output voltage (V_{OL}) is 0V
- the logic-1 input voltage (V_{IH}) is equal to or greater than 0.7×5V which is 3.5V
- the logic-0 input voltage (V_{IL}) is equal to or less than 0.3×5V which is 1.5V

These input and output voltages for 4000 series CMOS operating on 5V are illustrated in the diagram in Figure 13 on the next page.

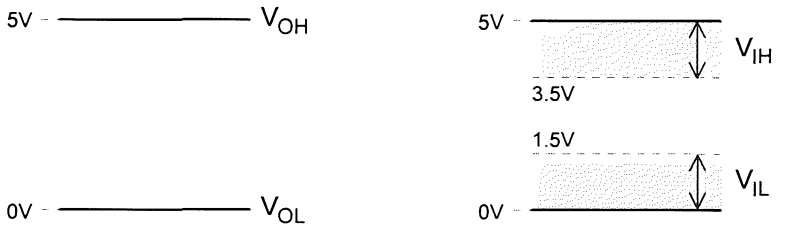


Figure 13 *The input and output voltages of 4000 series CMOS operating on a V_{DD} of 5V*

Practice calculating the input and output voltages for the logic-1 and logic-0 of 4000 series CMOS for yourself by trying the following questions.

1. If a 4000 series CMOS device is operating on a supply voltage (V_{DD}) of 14V, calculate the following:

   a) V_{OH} _____

   b) V_{OL} _____

   c) V_{IH} _____

   d) V_{IL} _____

4000 series CMOS voltage and current specifications

The current specifications for logic-1 and logic-0 for CMOS varies depending on the supply voltage. Typical values at 5V are shown in Table 8 below. You're not expected to be able to recall these specifications. The current specifications of 74ALS series TTL are also shown for comparison.

Table 8

Current specifications	4000 series	74ALS series
Logic-1 output current - I_{OH} (max)	1mA	0.4mA
Logic-0 output current - I_{OL} (max)	-1mA	-8mA
Logic-1 input current - I_{IH} (max)	10μA	20μA
Logic-0 input current - I_{IL} (max)	-10μA	-0.4mA

Advantages of TTL and CMOS

In discussing some of performance characteristics of TTL and 4000 series CMOS we have already touched on most of their advantages and disadvantages. The following is a summary:

TTL

- TTL is faster than CMOS
- TTL can be operated at higher frequencies
- TTL is not as susceptible to damage due to ESD as CMOS

CMOS

- CMOS can operate on a wider range of supply voltages than TTL
- CMOS uses less power than TTL
- CMOS has a greater fanout than TTL

TTL and 4000 series CMOS incompatibility

Figure 14a below shows a TTL logic gate driving a CMOS logic gate and Figure 14b shows the opposite.

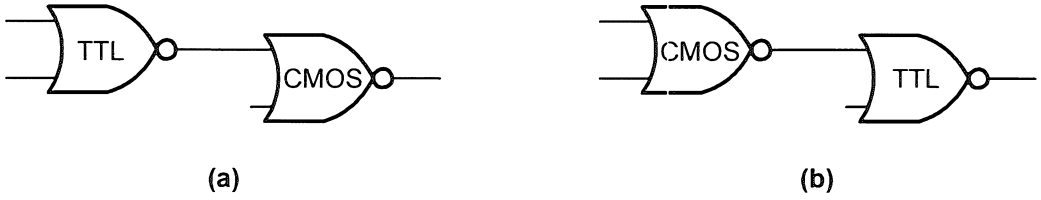


Figure 14 *Circuits involving logic devices from different logic families*

At first, it is reasonable to think that both of these circuits will work without any problem. However, neither circuit is likely to work reliably. The TTL and 4000 series CMOS logic families are not compatible and there are several reasons for this.

First, if the supply voltage for CMOS is anything other than 5V (remember, V_{DD} can be anything from 3V to 18V) the input and output voltages for logic-1 and logic-0 (that is, V_{OH}, V_{OL}, V_{IH} and V_{IL}) will be different to TTL. So, some logic-1s may be interpreted as logic-0 and vice versa. At the same time, some logic-1s and logic-0s may fall into the driven device's transition zone in which case the outputs will be indeterminate.

Second, even if the supply voltage for CMOS is 5V, it is still not possible for TTL devices to reliably drive CMOS because the TTL output voltage for a logic-1 (V_{OH}) can fall below the CMOS device's minimum allowable input for a logic-1 (V_{IH}). To explain, according to the data in Table 7 (on page 5-17), the minimum allowable V_{IH} for the 4000 series CMOS operating on a supply voltage of 5V is 3.5V. But the data in Table 3 (on page 5-8) shows that the minimum allowable V_{OH} of the TTL sub-families ranges from 2.4V to 2.7V. All of these are below 3.5V.

Third, even if the supply voltage for CMOS is 5V, CMOS devices would not be able to reliably drive most TTL devices because the sinking current (I_{OL}) of CMOS is below the current out of most TTL inputs when they are logic-0 (I_{IL}). To explain, the data in Table 8 (on page 5-18) shows that the maximum I_{OL} for the 4000 series CMOS operating on a supply voltage of 5V is about 1mA. But the data in Table 4 (on page 5-11) shows that the inputs of the 74, 74H, 74S and 74AS series TTL sub-families output currents from 1.6mA to 2mA. All of these are above 1mA.

Interconnecting TTL and 4000 series CMOS logic families

Despite the incompatibility of TTL and CMOS logic families, these devices can be still be used in circuits together but additional circuitry is needed to allow them to "talk" to each other.

To allow TTL devices to reliably drive 4000 series CMOS devices the circuit must prevent the output voltage for a logic-1 (V_{OH}) from dropping below the CMOS device's minimum allowable input for a logic-1 (V_{IH}). A simple resistor connected between the TTL device's output and the 5V supply rail will do this. The resistor is known as a *pull-up* resistor because it pulls the TTL's output voltage up. The pull-up resistor's value is typically between 1kΩ and 10kΩ.

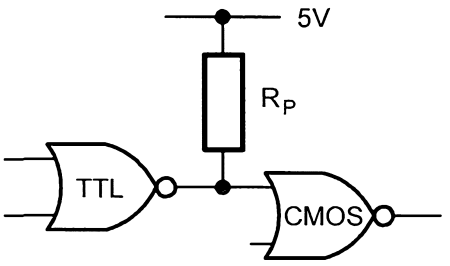


Figure 15 *TTL devices can reliably drive CMOS devices if the circuit includes a pull-up resistor*

CMOS devices can reliably drive up to two 74L, 74LS and 74ALS TTL inputs directly. To allow CMOS devices to drive the other TTL sub-families a special CMOS gate called a *buffer* must be connected between the two devices. There are buffers inside the 4050 and they have been designed specifically for this purpose. The 4049 is an IC that contains six inverting buffers that could also be used for this purpose.

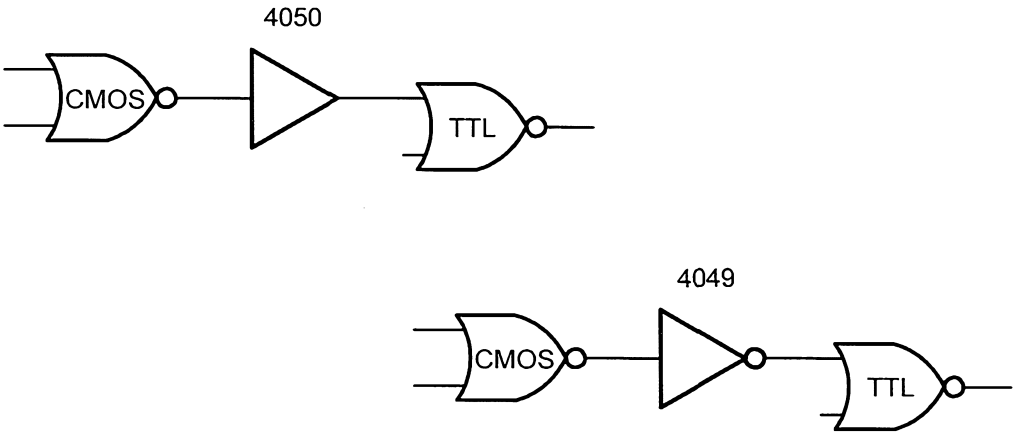


Figure 16 *The 4050 buffer and 4049 inverting buffer is used so that CMOS devices can drive the 74, 74H, 74S and 74AS TTL* sub-families

The TTL and 4000 series CMOS interface requirements are summarised in Figure 17 below (assuming all devices operated on a supply voltage of 5V).

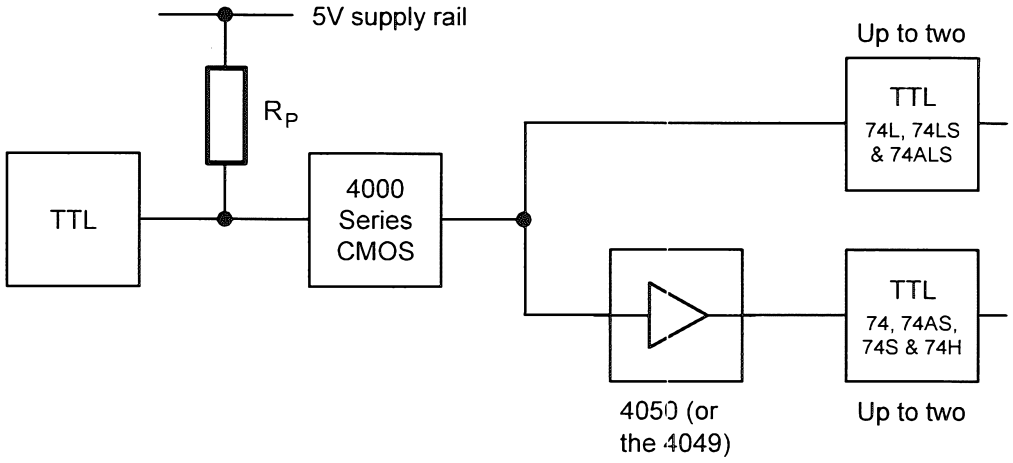


Figure 17 *Summary of the TTL and 4000 series CMOS interface requirements*

It is also possible to interface TTL logic devices to 4000 series CMOS devices operating on power supply voltages other than 5V and this will be dealt in a later section.

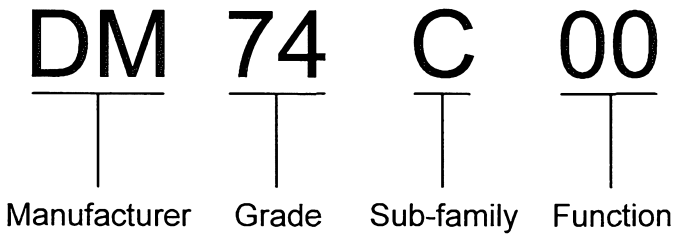
CMOS sub-families

As just discussed, there is little compatibility between 4000 series CMOS logic devices and TTL. The function numbers of 4000 series CMOS are not the same as for TTL and their voltage and current specifications for logic-1 and logic-0 are different so they cannot talk to each other without interfacing circuit. To overcome these problems chip manufacturers have developed CMOS sub-families with some or almost total compatibility with TTL.

To emphasise the compatibility of these CMOS devices with TTL, manufacturers use the 74 series part number system for these CMOS sub-families. An example of this is shown below.

DM74C00

Broken down, the part number reads as shown below.



Notice that this part numbering system is similar to the system for TTL. Moreover, unlike 4000 series CMOS devices, there are military grade versions of these CMOS sub-families.

At first, you might think that using the 74 series part number system for CMOS devices creates a potential for confusion: If a both TTL and CMOS devices use 74 then how can we tell them apart? The answer to this is actually simple. None of the TTL sub-families use the letter C between the grade number and the function number whereas all but one of these CMOS sub-families do. Examples include: 74C00, 74HC00, 74ACT00 and others (the exception is the 74LV series).

Technicians are expected to be able to work out whether an IC is CMOS or TTL and determine its sub-family. However, technicians would normally need data to determine the manufacturer of an IC and/or its function.

Below is the list of some of the commercially available 74 series CMOS sub-families and a brief description of their compatibility with TTL.

- 74**C**

  The 74C family is basically the 4000 CMOS series except the function numbers have been changed to correspond with TTL function numbers. So, the 74C00 has the same function as the 7400, the 74C32 has the same function as the 74AS32 and so on.

- 74**HC**

  The 74HC series is a high speed version of the 74C series and has operating speeds similar to 74LS series TTL. They also have TTL compatible outputs. This means that they can drive at least one input for all the TTL sub-families without the need for interfacing circuitry (the buffers are built-in). The 74HC series has a smaller range of operating voltages compared to 4000 series CMOS (2V to 6V instead of 3V to 18V).

- 74**HCT**

  The 74HCT series has almost complete compatibility with all TTL sub-families. That is, the function numbers are the same, the power supply requirements are the same (4.5V to 5.5V) and the inputs and outputs are TTL compatible. This means that this sub-family needs no interfacing circuitry when driving, or being driven by, TTL.

- 74**AC**

  The A is for *advanced* technology. This series has the same attributes as the 74HC series but is faster.

- 74**ACT**

  This series has the same attributes as the 74HCT series but is faster.

- 74**LV**

  The LV is for *low voltage*. This series has the same attributes as the 74HC series but can operate on much lower power supply voltages.

A quick survey by your author (in 2024) of several of the global component suppliers suggests that the 74HC series is the best selling of these families based on their price and suppliers' stock levels. Moreover, it would appear that the 74HC, 74HCT, 74AC and 74ACT all outsell the original TTL sub-families nowadays too.

Skill practice 5

Practise testing and measuring mixed logic family combinational logic circuits

This exercise is practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 45 minutes to complete this exercise.

Equipment

- Digital trainer
- DMM
- logic probe
- 7400
- 74LS08
- 4011
- 2k2Ω ¼W resistor
- hook-up wire (terminated)
- IC extractor

Remember:

Follow TAFE NSW WHS guidance at all times!

 Copyright: Not to be reproduced without permission

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

Part A - Setting up

2. Gather the equipment needed for this exercise.

3. Read the following points and make sure that you perform them when wiring all circuits:

   - Write the IC pin numbers of the gates you work with on the circuit that you're instructed to wire. Remember, pinout data for these ICs is at the back of the book.
   - Handle all ICs carefully - remember ESD can damage them.
   - Ensure the power supply is connected to all ICs in the circuit.
   - When inserting hook-up wire into the breadboard to make connections to the IC's pins, use holes that are furthest away from the pin.

4. Wire the circuit of Figure 1 below. **Note:** Use +5V for the supply to both chips.

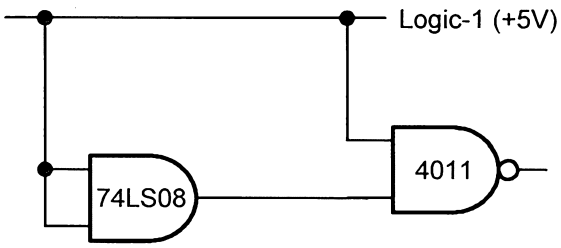


Figure 1

5. Measure the **TTL** logic gate's output voltage. Record your measurement in Table 1 below.

Table 1

Terminal characteristic	Measurement at Step 5	Measurement at Step 7
V_{OH}		

6. Modify the circuit as shown in Figure 2 below.

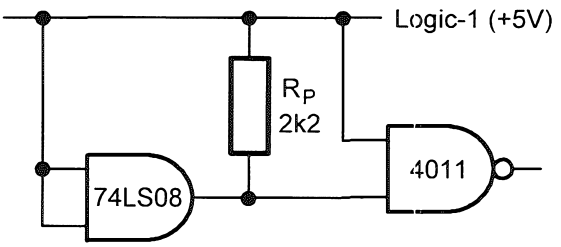


Figure 2

7. Measure and record (in Table 1 on the previous page) the TTL logic gate's new output voltage.

Question 1
What was the effect of adding the pull-up resistor to the circuit?

_____

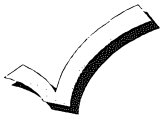
_____

Question 2
What problem does the pull-up resistor help to prevent when driving a CMOS input from a TTL output?

_____

_____

_____



The teacher needs to check your work at this point...

8. Set up the DMM for measuring currents using the **mA** range.

9. Wire the circuit of Figure 3 below. **Note:** Use +5V for the supply to both chips.

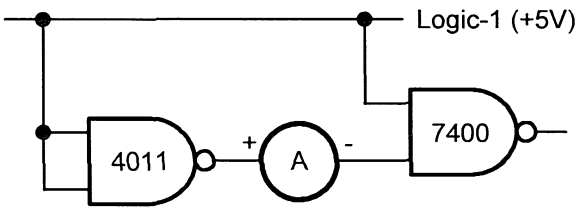


Figure 3

10. Measure the current flow between the CMOS and TTL logic gates. Record your measurement in Table 2 below.

Table 2

Terminal characteristic	Step 6
I_{OL}	

Question 3
According to the meter, in which direction is the current flowing?

☐ 4011 to 7400

☐ 7400 to 4011

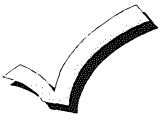
Question 4
When current flows in this direction, the CMOS output is

☐ sourcing current.

☐ sinking current.

Question 5
Why can the current out of the TTL logic gate's input be a problem for the CMOS logic gate's output?

Question 6
How can this problem be prevented?



The teacher needs to check your work at this point...

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for assessments.

1. Specify the grade, sub-family and family of the following ICs (the first one has been done for you as an example):

   a) 74LS32 Commercial grade low power schottky TTL

   b) 54193

   c) 4011

   d) 54ALS74

   e) 74HC04

   f) 74AS244

   g) 54ACT86

2. What is inside a logic IC if it is described as being a *dual four-input NAND gate*?

3. What is inside a logic IC if it is described as being a *hex inverter*?

4. What is an advantage of 4000 series CMOS over all TTL?

5. What is an advantage of all TTL over 4000 series CMOS?

 Copyright: Not to be reproduced without permission

6. What are the minimum and maximum allowable power supply voltages for the following TTL and CMOS devices? **Note:** This information is provided throughout the chapter.

a) 74LS32

b) 54193

c) 4011

d) 74HC40

e) 74HCT10

7. State two differences between 54 series (military grade) TTL and 74 series (commercial grade) TTL.

8. The reaction time of a logic device is known as the

- ☐ maximum clock frequency.
- ☐ propagation delay.
- ☐ fanout.
- ☐ power dissipation.

9. The number of logic gate inputs a logic gate output can drive is known as the

- ☐ maximum clock frequency.
- ☐ propagation delay.
- ☐ fanout.
- ☐ power dissipation.

10. What is *noise margin*?

11. Draw what you would expect to see at the output the gate in Figure 1 below for the inputs shown if the squarewave exceeded the device's maximum clock frequency.

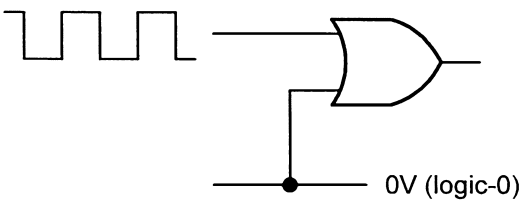


Figure 1

12. Which of the following ICs are TTL devices?

- ☐ 7400
- ☐ 74C32
- ☐ 74AS121
- ☐ 74ACT86
- ☐ 74ALS04
- ☐ 54AS74
- ☐ 4011
- ☐ 74HC08
- ☐ 74LS193

13. Which of the following ICs are CMOS devices?

- ☐ 7400
- ☐ 74C32
- ☐ 74AS121
- ☐ 74ACT86
- ☐ 74ALS04
- ☐ 54AS74
- ☐ 4011
- ☐ 74HC08
- ☐ 74LS193

14. Complete Table 1 below. See if you can do this from memory as you're expected to know these values.

Table 1

Device	V_{OH}	V_{OL}	V_{IH}	V_{IL}
74ALS series TTL				
4000 series CMOS				

Questions 15 to 24 refer to the data in Table 2

Table 2

Performance characteristic	74	74LS	74AS	74ALS	4000
Propagation delay (ns)	9	9.5	1.7	4	70
Power dissipation (mW)	10	2	8	1.2	0.01
Maximum clock rate (MHz)	35	45	200	70	5
Fanout	10	20	40	20	50

15. Name three specifications of 74AS devices that are better than the 74LS series.

16. Name the specification of 74ALS devices that is better than the 74AS series.

17. What specification (if any) of the original 74 devices is better than the other TTL devices?

18. What is the power dissipated by a 74AS00 IC? (**Note:** This IC is a *quad two-input NAND gate.*)

19. Name two specifications of 4000 CMOS devices logic that are better than all of the TTL devices.

_____

_____

20. Name two specifications of all of the TTL logic sub-families listed that are better than 4000 series CMOS logic.

_____

_____

21. Why is the 74ALS139 IC a suitable replacement for the 74LS139?

_____

_____

22. Why is a 74AS121 probably a suitable replacement for a 74121?

_____

_____

23. Why is a 74AS00 **not** a suitable replacement for a 4000?

_____

_____

24. Which of the following devices is the best substitute for a 74LS73?

- ☐ 74AS73
- ☐ 74ALS73
- ☐ 7473

Questions 25 to 35 refer to the data in Table 3

Table 3

Specification	74	74LS	74AS	74ALS	4000	74HCT
V_{OH} (min)	2.4	2.7	2.5	2.5	V_{DD}	3.84
V_{OL} (max)	0.4	0.5	0.5	0.4	0V	0.33
V_{IH} (min)	2.0	2.0	2.0	2.0	$0.7V_{DD}$	2.0
V_{IL} (max)	0.8	0.8	0.8	0.8	$0.3V_{DD}$	0.8
I_{OH} (max)	0.4	0.4	1.0	0.4	1	4
I_{OL} (max)	-16	-8	-20	-8	-1	-4
I_{IH} (max)	40μA	20μA	50μA	20μA	10μA	1μA
I_{IL} (max)	-1.6	-0.4	-2.0	-0.4	-10μA	-1μA

Note: All currents in milliamps unless otherwise specified.

25. What is the range of voltages for a logic-1 on the output of a 74LS02?

26. What is the range of voltages for a logic-0 on the output of a 74ALS08?

27. What is the range of allowable voltages for a logic-1 on the input of a 74AS76?

28. What is the range of allowable voltages for a logic-0 on the input of a 74ALS04?

29. What is the range of voltages for the transition zone of a 74LS00?

30. How much current goes into the input of a 74AS02 when it is a logic-1?

31. How much current comes out of the input of a 74LS86 when it is a logic-0?

32. What is the technical term for the current in or out of the inputs of logic device's?

33. How much current can the output of a 74ALS30 source?

34. How much current can the output of a 74AS32 sink?

35. What is the noise margin of the 74LS series TTL?

36. What is the noise margin of the 74AS series TTL?

37. Why can't 4000 series CMOS devices reliably drive most TTL sub-families?

38. What is needed to allow 4000 series CMOS devices to reliably drive TTL devices?

- ☐ A pull-up resistor
- ☐ A CMOS buffer

39. Why can't TTL devices reliably drive 4000 series CMOS devices with a V_{DD} of 5V?

40. What is needed to allow TTL devices to reliably drive 4000 series CMOS devices?

- ☐ A pull-up resistor
- ☐ A CMOS buffer

41. Which of the following CMOS devices is the most compatible with TTL?

- ☐ 74C
- ☐ 74AC
- ☐ 74HC
- ☐ 74HCT

Section 6 Number systems in digital electronics

Purpose To develop your ability to convert between the different number systems used in digital electronics.

Objectives At the end of this section you should be able to:

- Convert binary numbers to decimal and hexadecimal without the aid of a calculator.
- Convert decimal numbers to binary and hexadecimal without the aid of a calculator
- Convert hexadecimal numbers to binary and decimal without the aid of a calculator
- Use a logic probe to test and measure logic states on digital IC that work with binary numbers

Introduction

So far for this unit you've learnt how to count up in binary from 0000 to 1111 (recall, that that's 0 to 15 in decimal). However, many digital systems like personal computers, mobile phones, gaming consoles (like PS5) work with 64-bit binary numbers. That is, they work with numbers from 0000000000000000000000000000000000000000000000000000000000000000 to 1111111111111111111111111111111111111111111111111111111111111111. It's remarkable that up until about 20 years or so ago only *super-computers* could work with binary numbers so large.

However, as you know, people use the decimal number system and not the binary number system. That means that, as a technician, when repairing digital electronics equipment you may sometimes need to convert binary numbers into decimal and vice versa. You may also need to be able to convert binary and decimal numbers to another number system called *hexadecimal*. We'll learn about hexadecimal and how to perform these conversions in this section.

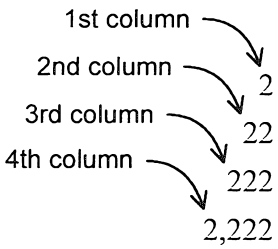
The principles of number systems

Most number systems are based on the same set of principles. So, to learn more about binary and hexadecimal and how to convert between them, it's useful to learn about these principles first. And, to help explain them, we'll use the number system that you're very familiar with, decimal.

As you know, decimal numbers that consist of more than one digit are built up from right to left. This is demonstrated below using four numbers.

2
22
222
2,222

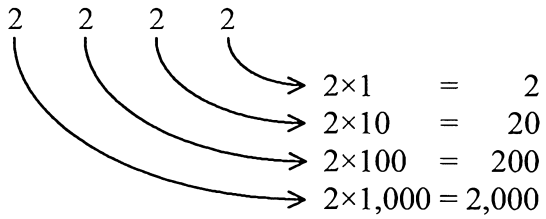
When these four numbers are written this way we can see that the individual digits that make up the numbers are arranged in columns. These are highlighted below.



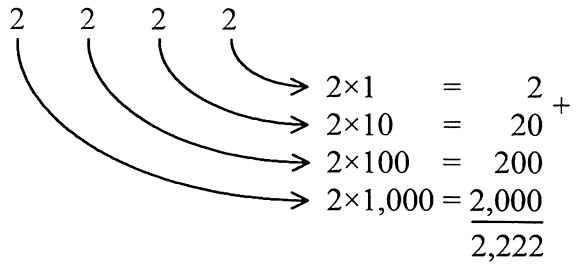
Importantly, each column is said to be *weighted.* That is, each column is worth a different fundamental amount. So, although the digit 2 appears in every column for the number 2,222, each 2 is worth a different amount depending on the column it's in.

The first column is worth ones or *units* (and this is true of all the number systems that we'll learn about in this module). So the total amount represented by the number 2 in this column is two lots of one (that is, 2×1) which is two.

The second column is worth tens so the total amount represented by the number 2 in this column is two lots of ten (2×10) which is twenty. The third column is worth hundreds so the total amount represented by the number 2 in this column is two lots of one hundred (2×100) which is two hundred. And finally, the fourth column is worth thousands so the total amount represented by the number 2 in this column is two lots of one thousand (2×1000) which is two thousand. This is summarised below.



Shown this way we can see that the digits that make up the number 2,222 (or any number for that matter) are really acting as *multipliers* (like the multiplier band in the resistor colour code) because each column's weighting is multiplied by the value of each digit in that column. When you add up the results of these multiplications you get the total amount which in this case is 2,222. This is shown below.



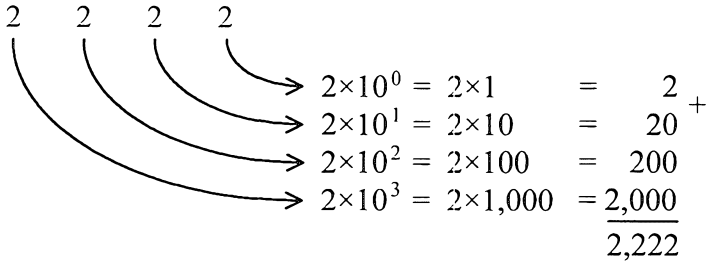
If all this seems pretty obvious, that's because you've grown up with decimal and so you think in decimal.

What you may not know about decimal is that the columns' weighting is determined mathematically. The first column is equal to ten raised to the power of zero (that is, 10^0 which is equal to 1). The second column is equal to ten raised to the power of one ($10^1 = 10$). The third column is equal to ten raised to the power of two ($10^2 = 100$) and so on. This is summarised in Table 1 below for four columns. As ten is the base of the mathematical operations used to calculate each column's weighting, decimal is also known as *Base-10*.

Table 1

		4th column	3rd column	2nd column	1st column
Decimal	Weighting expressed as ten raised to a power	10^3	10^2	10^1	10^0
	Weighting as a base number	1,000	100	10	1

Using this information we can rewrite the mathematical summary discussed on the previous page as shown below.



In the end, the total is the same but knowing how the weighting of the columns in decimal are determined can help us to understand how the binary and hexadecimal number systems work and how to convert between them.

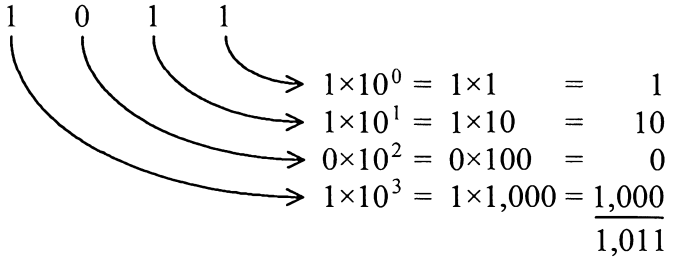
Binary

As you know, binary is a number system that consists of only two digits (zero and one) and hence its name. But binary is also known as *Base-2* because the weighting of each column of a binary number is determined mathematically by raising two (instead of ten) to a power. Table 2 below shows a comparison of the weighting of four columns in binary (and decimal for comparison).

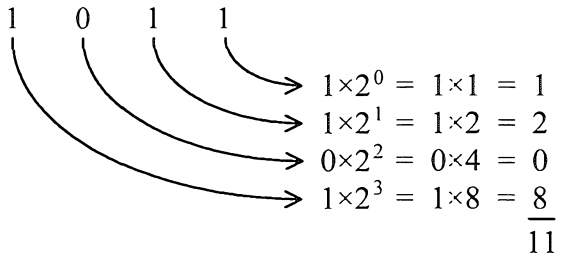
Table 2		4th column	3rd column	2nd column	1st column
Decimal	Weighting expressed as ten raised to a power	10^3	10^2	10^1	10^0
	Weighting as a base number	1,000	100	10	1
Binary	Weighting expressed as two raised to a power	2^3	2^2	2^1	2^0
	Weighting as a base number	8	4	2	1

Notice that the first column is worth one in both the decimal and binary number systems. Notice also that the weighting of each column in binary, expressed as a base number, is double the weighting of the column to its right. This is an attribute of binary that is very handy to remember.

Importantly, as the weighting of the columns in binary is different to that of decimal, numbers written in binary refer to a different quantity as the same numbers written in decimal. For example, let's say you have 1,011 oranges. If this number of oranges is a decimal quantity then you'd have 1,011 of them. Proof:



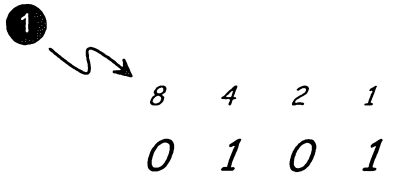
However, if this number of oranges is a binary quantity then you'd have only 11 of them when counted using decimal. Proof:



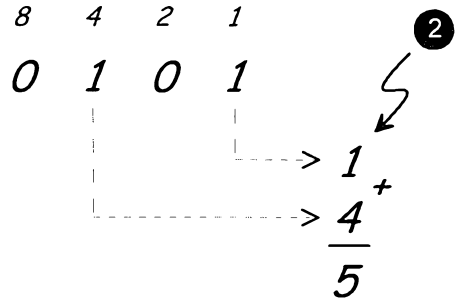
Converting binary numbers to decimal

The previous discussion forms the basis of the surprisingly simple procedure for converting binary numbers into decimal. The procedure is described below with each step being demonstrated using the example of converting the binary number 0101 to decimal.

1. Write the weight of the column (as a base number) for every digit of the binary number being converted. This is usually written across the top of the binary number.



2. Add the weighted value of all columns with a one in them. The total is the decimal equivalent.



This example demonstrates that 5 is the decimal equivalent of the binary number 0101.

Practise converting binary numbers to decimal for yourself by trying the following examples.

1. 0111

2. 1010

3. 1100

4. 1111

5. 01011

6. 11001

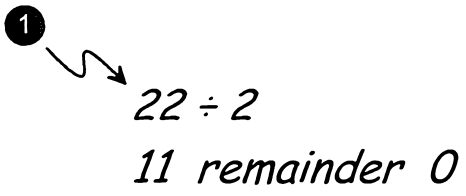
7. 01010001

8. 11111101

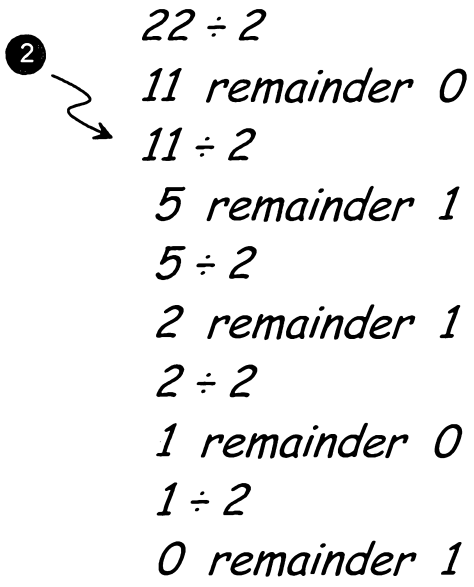
Converting decimal numbers to binary

There are two methods for converting decimal numbers to binary. The procedure for the first is described below with each step being demonstrated using the example of converting the decimal number 22 to binary.

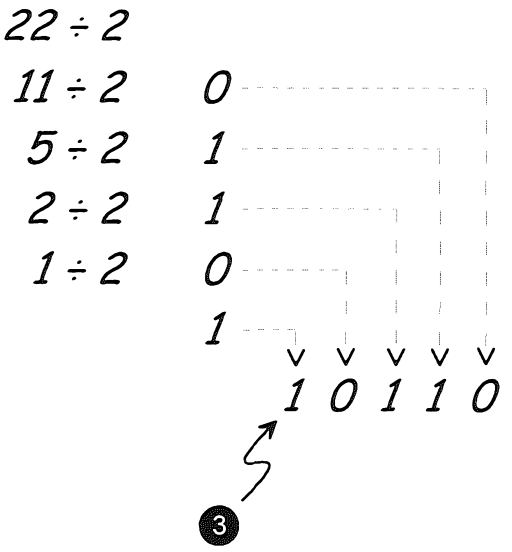
1. Divide the decimal number by two and do two things: a) write down the answer below it; and b) write the remainder (which will be either a zero or a one) next to it.



2. Repeat the first step on the answer to the first division and all successive answers until you get zero.



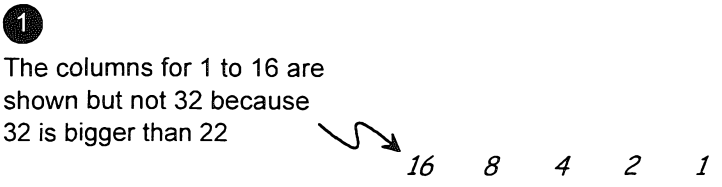
3. Then, the zeros and ones in the remainder column form the binary equivalent. The remainder of the first division is the least significant bit and the remainder of the last division is the most significant bit.



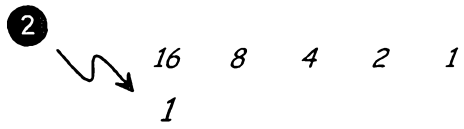
This example demonstrates that 10110 is the binary equivalent of the decimal number 22. If the digital system that you're working with uses 8-bit binary words then 10110 is written as 00010110. Although the two numbers look different, their magnitude is the same (this can be proven if you convert both numbers back to decimal).

The second method of converting decimal numbers to binary is based on a process known as *successive approximation*. The procedure is described below with each step being demonstrated using the same example of converting the decimal number 22 to binary.

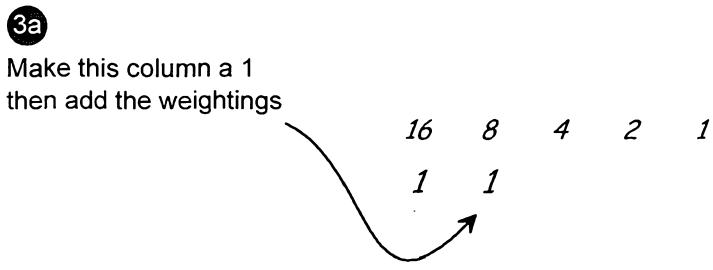
1. Write the binary weighting for the columns starting from the right but stop at the first column with a weighting bigger than the number you want to convert. For example:

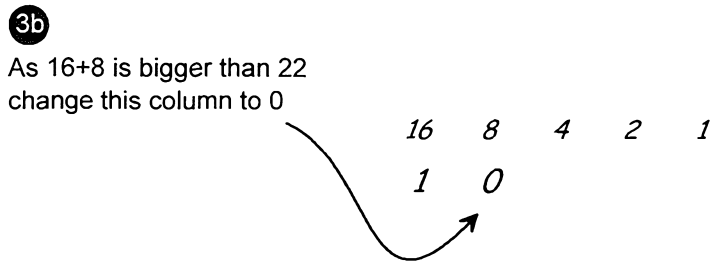


2. Put a "1" in the first column as shown below.

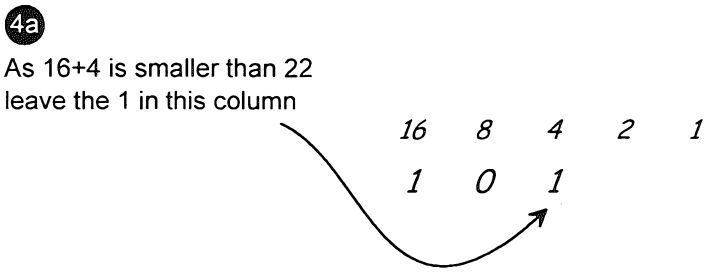


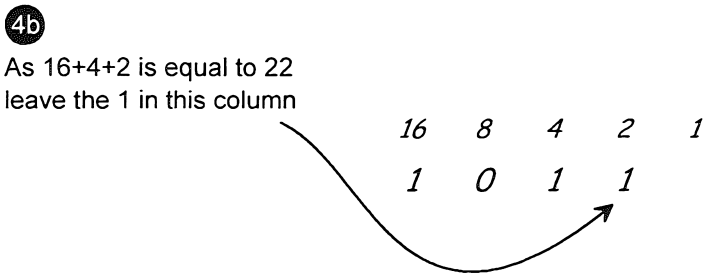
3. Put a "1" in the next column then add the two columns' weighting. If the total is smaller than or equal to the decimal number that your converting leave the column as "1". If it's bigger, change the column to "0". For example:



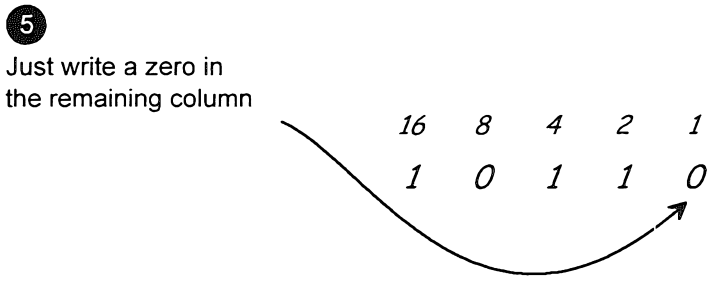


4. Repeat Step 3 for every column, one at a time, working from left to right. Importantly, you must add the weighting of all preceding columns with a "1" in them to decide whether the current column remains a "1" or should be changed to a "0".





5. Sooner or later you'll find that the sum of the weighted columns with a "1" in them is equal to the decimal number being converted. When this happens, simply write "0" in the remaining columns. For example:



That said, the two numbers may not be equal until you put a "1" in the final column. This is fine and the conversion is complete.

Practise converting decimal numbers to either 4-bit or 8-bit binary numbers for yourself by trying the following examples. Remember, you can always check your answer by converting them back to decimal. If it is the same as the number you started out with then your answer is probably correct.

1. 7

2. 13

3. 15

4. 16

5. 56

6. 127

7. 128

8. 255

On a final note, you can probably imagine that there is the potential for confusion when working with numbers with different bases. For example, the number 1,010 could be either a decimal or a binary number. There is nothing to tell you whether it is one or the other.

There isn't a common method used to identify numbers as being either decimal or binary in general electronics. This is probably because experience is usually enough to avoid confusion. As a tip though, longer decimal numbers tend to be grouped in lots of three while longer binary numbers are usually grouped in lots of four or eight. So, the number 1 010 is probably decimal while the number 1010 is probably binary. Similarly, the number 11 011 010 is probably decimal while the numbers 1101 1010 and 11011010 are probably binary.

To make your life a little easier while studying this subject, your author uses a method of denoting the base of a number that is commonly used in general mathematics. Unless specified in words, decimal numbers will have the number ten written at the end of it in subscript: for example, 1010_{10}. A binary number will have the number two written at the end of it in subscript: for example, 1010_2.

Hexadecimal

For reasons that we'll explain later in this section, *hexadecimal* is another number system commonly used in digital electronics. There is a significant difference between hexadecimal and decimal - hexadecimal has sixteen counts in the count sequence before it repeats (hence the name is made up of the words hex and decimal) whereas decimal has only ten.

For every count in the single-digit sequence of any number system there must be a digit to represent the count. For example, binary has two counts and therefore two digits (zero and one). Decimal has ten counts and therefore ten digits (zero through nine). As hexadecimal has sixteen counts it must have sixteen digits.

This might seem like a problem at first because, as you know, there are only ten available numbers (0 through to 9). However, as numbers are really just signs that represent a quantity, all that is needed are another six signs. And, the first six letters of the English alphabet (A to F) are as good as any other signs.

Table 3 below shows the digits in the hexadecimal count sequence and the corresponding decimal and binary numbers.

Table 3

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

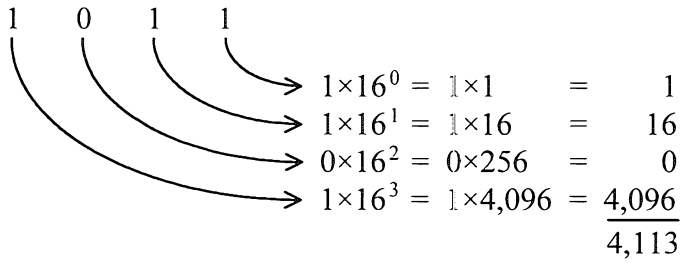
To make hexadecimal numbers bigger than F, use the same procedure for making numbers bigger than nine in decimal or bigger than one in binary. That is, reset the column and add one to the left. For example, the hexadecimal number that is one bigger than F is 10 (which is really 16_{10}). The hexadecimal number that is two bigger than F is 11 (which is really 17_{10}) and so on.

Hexadecimal is also known as *Base-16* because the weighting of each column of a hexadecimal number is determined mathematically by raising sixteen to a power. Table 4 below shows a comparison of the weighting of four columns in hexadecimal (and decimal for comparison). Notice that the first column is worth one in both systems.

Table 4

		4th column	3rd column	2nd column	1st column
Decimal	Weighting expressed as ten raised to a power	10^3	10^2	10^1	10^0
	Weighting as a base number	1,000	100	10	1
Hexa-decimal	Weighting expressed as sixteen raised to a power	16^3	16^2	16^1	16^0
	Weighting as a base number	4096	256	16	1

Now, if you have 1,011 oranges and this number is a hexadecimal quantity then you'd really have $4{,}113_{10}$ of them. Proof:



There are four methods of denoting numbers as hexadecimal values. In general mathematics, the number sixteen is written at the end of the number in subscript form: for example, 1011_{16}. However, in digital electronics and computing three other methods are preferred. The first involves writing the letter aitch ("H" or "h") at the end of the number: for example, 1011H or 1011h. The second involves writing the dollar sign in front of the number: for example, $1011. The third involves writing 0x in front of the number: for example, 0x1011. This workbook standardises on the dollar sign.

Why is hexadecimal used?

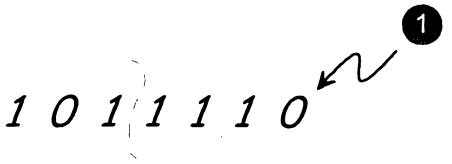
As you know, digital electronics equipment work with zeros and ones. So any numbers that the equipment works with must be binary numbers. However, as already mentioned, some digital electronics equipment work with thirty two bit numbers and more. The length of these numbers and the fact that they are made up of only zeros and ones makes them difficult for people to read and work with.

For most people, it is easier to convert these binary numbers to a number system with a higher base first before working with them. This way, the numbers will be smaller and easier to read. However, engineers and technicians usually convert binary numbers to hexadecimal (Base-16) rather than decimal (Base-10). The reason for this is simple, the conversion from binary to hexadecimal and back again is easier (both for people and for digital circuits).

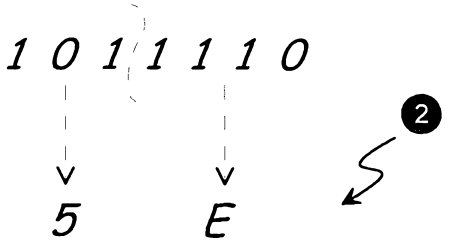
Converting binary numbers to hexadecimal

The procedure to convert hexadecimal numbers to binary is described below with each step being demonstrated using the example of converting the number 1011110_2 to hexadecimal.

1. Starting from the least significant bit of the binary number to be converted, group the bits in lots of four.



2. Convert each group of four bits into its equivalent hexadecimal digit then put the digits together.

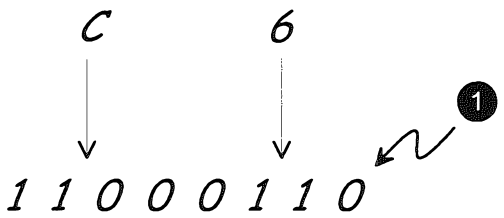


This example demonstrates that \$5E is the hexadecimal equivalent of the number 1011110_2.

Converting hexadecimal numbers to binary

The procedure to convert binary numbers to hexadecimal is described below with the step being demonstrated using the example of converting the number $C6 to binary.

1. Convert each hexadecimal digit into its equivalent 4-bit binary number then put the 4-bit binary numbers together.



This example demonstrates that 11000110 is the binary equivalent of the number $C6.

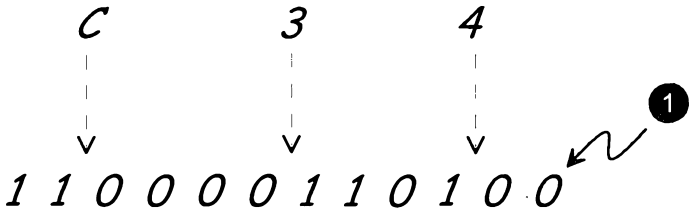
Practise converting between hexadecimal and binary numbers for yourself by trying the following questions.

1. Convert 01101000_2 to hexadecimal.
2. Convert $33 to binary.
3. Convert 10111101_2 to hexadecimal.
4. Convert $1AE to binary.

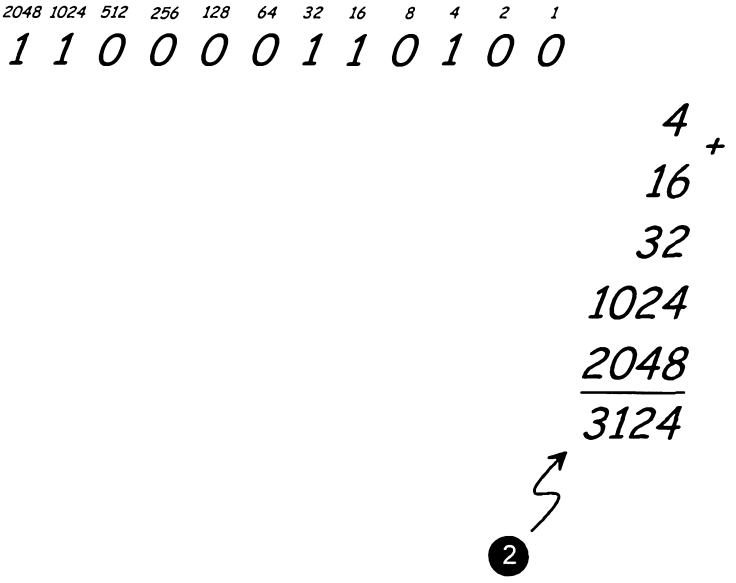
Converting hexadecimal numbers to decimal

Although engineers and technicians usually work in hexadecimal, you'll sometimes need to convert a hexadecimal number to decimal. There are two methods for this and the procedure for the first is described below with each step being demonstrated using the example of converting the number $C34 to decimal.

1. Convert the hexadecimal number to binary.



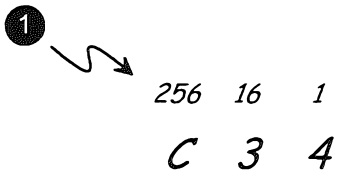
2. Convert the binary equivalent to decimal.



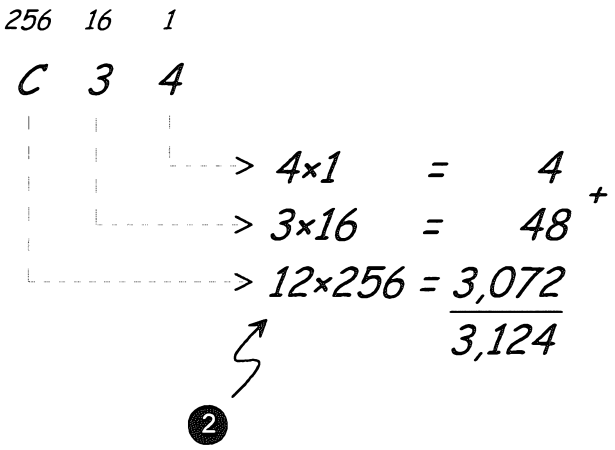
This example demonstrates that 3,124 is the decimal equivalent of the number $C34.

The procedure for the second method is described below with each step being demonstrated using the same example of converting the number $C34 to decimal.

1. Write the weight of the column (as a base number) for each digit of the hexadecimal number being converted. This is usually written across the top of the hexadecimal number.



2. Multiply the decimal equivalent of every hexadecimal digit by the weighted value of its column and add the results. The total is the decimal equivalent.



This second example also demonstrates that 3,124 is the decimal equivalent of the number $C34.

Practise converting hexadecimal numbers to decimal for yourself by trying the following questions.

1. $1F

2. $A4

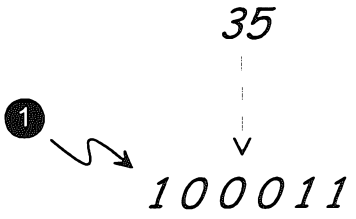
3. $3E2

4. $7FF

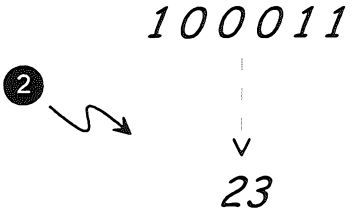
Converting decimal numbers to hexadecimal

There are two methods for converting decimal numbers to hexadecimal. The procedure for the first is described below with each step being demonstrated using the example of converting the number 35_{10} to hexadecimal.

1. Convert the decimal number to binary.



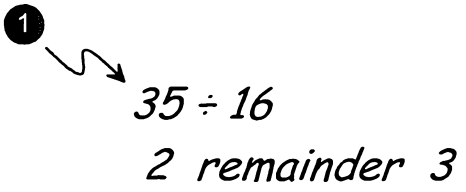
2. Convert the binary equivalent to hexadecimal.



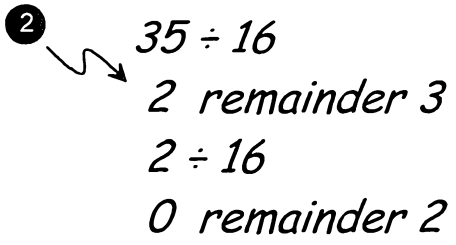
This example demonstrates that $23 is the hexadecimal equivalent of 35_{10}.

The procedure for the second method is described below with each step being demonstrated using the same example of converting the number 35_{10} to hexadecimal.

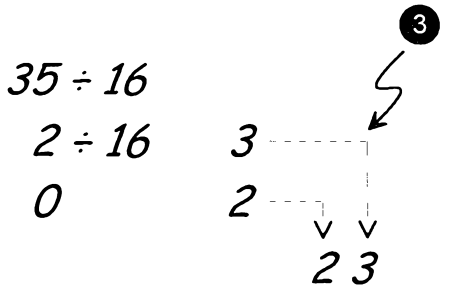
1. Divide the decimal number by sixteen and do two things: a) write down the answer below it; and b) write the remainder as a hexadecimal value next to it.



2. Repeat the first step on the answer and all successive answers until you get zero.



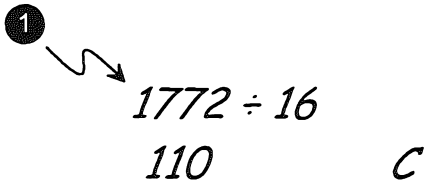
3. Then, the hexadecimal values in the remainder column form the hexadecimal equivalent. The remainder of the first division is the least significant bit and the remainder of the last division is the most significant bit.

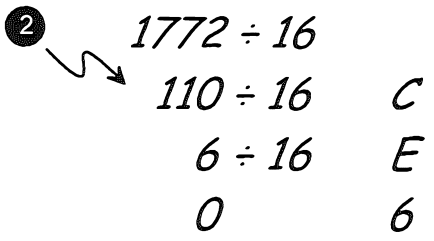


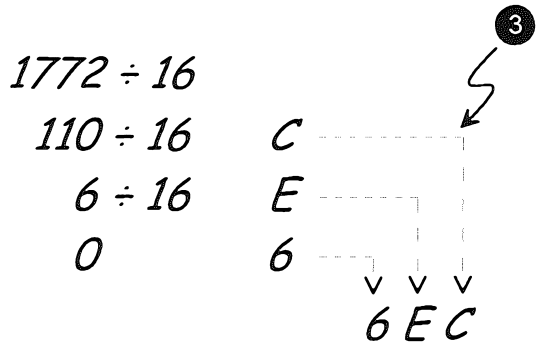
This second example also demonstrates that \$23 is the hexadecimal equivalent of 35_{10}.

This simple example misses an important aspect of decimal to hexadecimal conversions using this method. Namely, the remainder must be written as a hexadecimal value. For example, if the remainder 13, then you write D (which is the hexadecimal equivalent of 13) instead of 13.

Let's do another conversion to emphasise this point and convert the number $1,772_{10}$ to hexadecimal.







This example demonstrates that \$6EC is the hexadecimal equivalent of $1,772_{10}$.

Practise converting decimal numbers to hexadecimal for yourself by trying the following questions.

1. 44

2. 71

3. 211

4. 2,314

Copyright: Not to be reproduced without permission

Student notes

Skill practice 6

Practise testing and measuring digital ICs

This exercise is a practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 40 minutes to complete this exercise.

Equipment

- Digital trainer
- DMM
- logic probe
- 74LS283*
- hook-up wire (terminated)
- IC extractor

* The part number of the IC that you will be using may have letters after the 74 (for example, 74LS83 or 74H83). These letters can be ignored for the purpose of this practical.

Remember:

Follow TAFE NSW WHS guidance at all times!

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

2. Gather the equipment needed for this exercise.

3. Read the following points and make sure that you perform them when wiring all circuits:

   - Handle all ICs carefully - remember ESD can damage them.
   - When plugging hook-up wire into the breadboard to make connections to the pins of the IC, use holes that are furthest away from the IC's pin.

4. Wire the circuit of Figure 1.

   Note: The numbers inside the box in the diagram denote the pin connections to the IC. Their order has be chosen for convenience and does not represent the order of their physical arrangement of the IC's pin numbers.

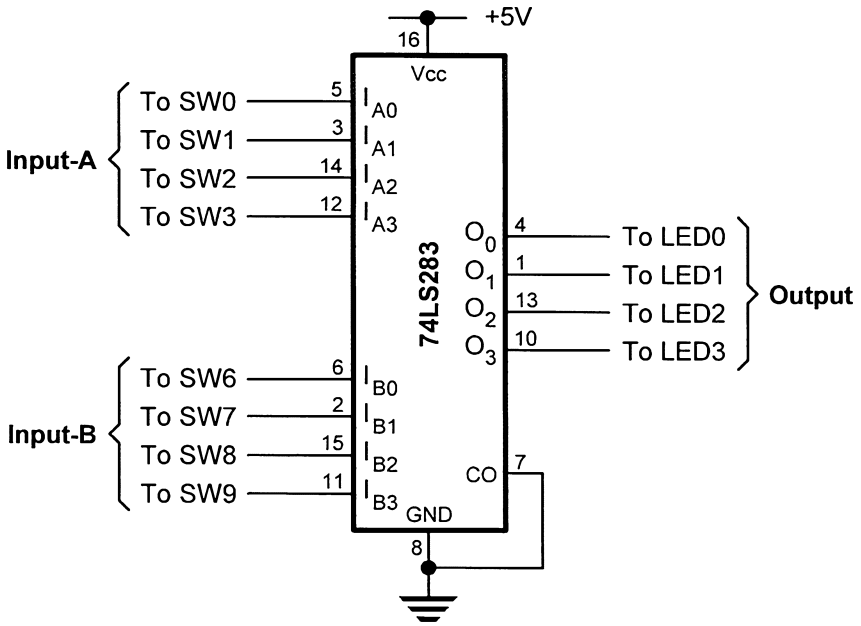


Figure 1

5. Turn on the digital trainer.

6. Set the toggle switches for input-A to 0011.

7. Set the toggle switches for input-B to 0100.

8. Record the binary value on the 74283's output displayed on the LEDs in Table 1 below.

9. Convert the three binary numbers to decimal.

Table 1

	Binary				Decimal
	I_3	I_2	I_1	I_0	
Input-A	0	0	1	1	
Input-B	0	1	0	0	
Output					

10. Set the toggle switches for input-A to 0101.

11. Set the toggle switches for input-B to 0111.

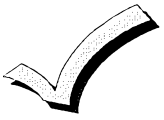
12. Record the binary value on the 74283's output in Table 2 below.

13. Convert the three binary numbers to decimal.

Table 2

	Binary				Decimal
	I_3	I_2	I_1	I_0	
Input-A	0	1	0	1	
Input-B	0	1	1	1	
Output					

Question 1
Compare the numbers on the inputs and output of the 74283 in both tables. What do you think is the function of the 74283 IC?



The teacher needs to check your work at this point...

14. Set the toggle switches for input-A to 1100.
15. Set the toggle switches for input-B to 1000.
16. Record the binary value on the 74283's output in Table 2 below.
17. Convert the three binary numbers to decimal.

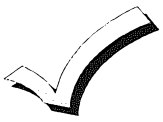
Table 3

	Binary				Decimal
	I_3	I_2	I_1	I_0	
Input-A	1	1	0	0	
Input-B	1	0	0	0	
Output					

Copyright: Not to be reproduced without permission

Question 2
What number (in decimal) should have appeared on the output of the 74283 IC?

Question 3
Why is the 74283's output incorrect? [**Tip:** If you're not sure why, convert your answer to question 2 above to binary. The answer should give you a clue.]



The teacher needs to check your work at this point...

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. Convert the following binary numbers to decimal.

a) 110

b) 0100

c) 1011

d) 1101

e) 00001001

f) 00010001

g) 01100110

h) 10011111

Copyright: Not to be reproduced without permission

2. Convert the following decimal numbers to 4-bit binary numbers.

a) 8

b) 14

3. Convert the following decimal numbers to 8-bit binary numbers.

a) 8

b) 14

c) 18

d) 42

e) 70

f) 101

g) 128

h) 248

4. Convert the following binary numbers to hexadecimal.

a) 1000

b) 1110

c) 00000011

d) 00001100

e) 00010001

f) 01100110

g) 10011111

h) 110100011010

5. Convert the following hexadecimal numbers to 4-bit binary numbers.

a) $2

b) $A

6. Convert the following hexadecimal numbers to 8-bit binary numbers.

a) $05

b) $0B

c) $36

d) $7F

e) $BD

f) $FF

7. Convert the following hexadecimal numbers to 12-bit binary numbers.

a) $6E2 b) $ABC

8. Convert the following hexadecimal numbers to 16-bit binary numbers.

a) $1FFF b) $2000

 Copyright: Not to be reproduced without permission

9. Convert the following hexadecimal numbers to decimal.

a) $15

b) $2C

c) $92

d) $B3

e) $0C1

f) $1AF

g) $400

h) $818

10. Convert the following decimal numbers to hexadecimal.

a) 29

b) 50

c) 107

d) 255

e) 256

f) 328

g) 411

h) 1,215

 Copyright: Not to be reproduced without permission

Student notes

Section 7 Codes in digital electronics

Purpose To develop your understanding of three codes commonly used in digital electronics equipment.

Objectives At the end of this section you should be able to:

- Convert decimal numbers to BCD without the aid of a calculator
- Convert BCD to decimal numbers without the aid of a calculator
- List the advantages and disadvantages of BCD compared to binary
- Give examples of digital electronics equipment that use BCD
- Convert alphanumeric/control characters to ASCII
- Convert ASCII to alphanumeric/control characters
- Explain the purpose of control characters in ASCII
- Explain the basic differences between ASCII and Unicode
- Give applications in which Unicode is used
- Explain the difference between Gray code and binary
- State an advantage and a disadvantages of Gray code compared to binary
- Give examples of digital electronics devices and equipment that use Gray code

Introduction

So far, you have learnt that ones and zeros can be put together to form binary numbers. However, it would be problematic to use only the binary number system in many types of digital equipment. To explain, being a number system, binary obviously doesn't have letters or other things like punctuation marks. This isn't much good for computers or mobile phones. Second, complex circuitry is needed to convert binary numbers into decimal numbers for display (like on a clock radio).

For this reason, many digital devices use digital's two logic levels to make codes instead of, or in addition to, binary. That is, ones and zeros are put together to form codes to represent numbers, letters, punctuation marks and other things it much the same way that Morse code uses dots and dashes. There are many digital codes but we'll consider just three here: *binary coded decimal*, *ASCII* and *Gray code*.

Binary coded decimal

Binary coded decimal (BCD) is a code that is used to represent the decimal digits 0 to 9. Each decimal digit is represented by a 4-bit code that is the binary equivalent of the decimal digit. This is shown below in Table 1.

Table 1

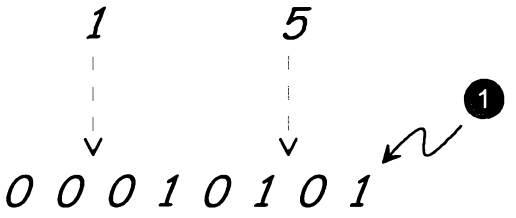
Decimal digits	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Importantly, there are only ten unique codes in BCD. So, when decimal numbers bigger than nine are represented in BCD, every decimal digit in the number must be represented by its own 4-bit code.

Converting decimal numbers to BCD

The procedure to convert decimal numbers to BCD is described below with the step being demonstrated using the example of converting the number 15_{10} to BCD.

1. Convert each decimal digit into its equivalent 4-bit binary number then put the 4-bit binary numbers together.



This example demonstrates that 0001 0101 is the BCD equivalent of the decimal number 15.

On first appearances, binary coded decimal (BCD) looks just like binary. However, we can see that there is a significant difference between the two when we compare decimal numbers greater than nine converted to both systems. Table 2 below shows some examples.

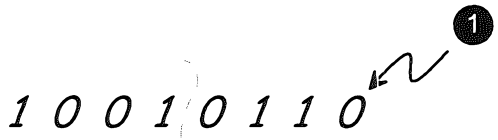
Decimal	BCD	Binary
10	0001 0000	1010
15	0001 0101	1111
39	0011 1001	100111
162	0001 0110 0010	10100010

As you can see, numbers bigger than nine in BCD always consist of more bits (that is, ones and zeros) than the same numbers in binary.

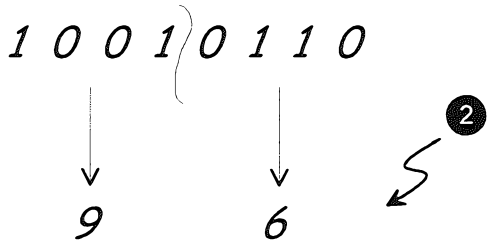
Converting BCD numbers to decimal

The procedure to convert BCD to decimal is described below with each step being demonstrated using the example of converting the code 1001 0110 to decimal.

1. Starting from the least significant bit of the binary number to be converted, group the bits in lots of four.



2. Convert each group of four bits into its equivalent decimal digit then put the digits together.



This example demonstrates that 96 is the decimal equivalent of the BCD 1001 0110.

Practise converting decimal numbers to BCD and vice versa for yourself by trying the following questions.

1. Convert 53_{10} to BCD.

2. Convert 3268_{10} to BCD.

3. Convert 1001 0111 to decimal.

4. Convert 0001 0100 1000 to decimal.

Applications for BCD

It is easier to convert between decimal and BCD numbers than it is to convert between decimal and binary numbers. This is true for both people and digital circuits. That being the case, BCD finds applications in digital electronics where systems must input and/or output numbers in decimal form. Examples include digital clocks, the keypad for alarm systems, digital speedometers in cars, the display of digital multimeters and so on.

However, although it is easier to convert between decimal and BCD, BCD has its limitations. First, the circuits required to perform mathematical operations on BCD numbers are much more complex than circuits designed to perform mathematical operations on true binary numbers. So, binary is preferred in computer systems and calculators instead of BCD.

Another limitation of BCD is there are only ten codes (0000 to 1001). This means it cannot be used to represent letters in the English language alphabet. So BCD cannot be used in applications such as wordprocessing or text messaging and so on.

ASCII

The *American Standard Code for Information Interchange* (ASCII) was developed in the mid 1960s to represent numbers, letters and punctuation marks. There are 128 unique ASCII codes and they are shown below in Table 3 along with the character or action that each code represents.

Table 3

Character	ASCII	Character	ASCII	Character	ASCII	Character	ASCII
NUL	0000000	Space	0100000	@	1000000	`	1100000
SOH	0000001	!	0100001	A	1000001	a	1100001
STX	0000010	"	0100010	B	1000010	b	1100010
ETX	0000011	#	0100011	C	1000011	c	1100011
EOT	0000100	$	0100100	D	1000100	d	1100100
ENQ	0000101	%	0100101	E	1000101	e	1100101
ACK	0000110	&	0100110	F	1000110	f	1100110
BEL	0000111	'	0100111	G	1000111	g	1100111
BS	0001000	(	0101000	H	1001000	h	1101000
HT	0001001	)	0101001	I	1001001	i	1101001
LF	0001010	*	0101010	J	1001010	j	1101010
VT	0001011	+	0101011	K	1001011	k	1101011
FF	0001100	,	0101100	L	1001100	l	1101100
CR	0001101	-	0101101	M	1001101	m	1101101
SO	0001110	.	0101110	N	1001110	n	1101110
SI	0001111	/	0101111	O	1001111	o	1101111
DLE	0010000	0	0110000	P	1010000	p	1110000
DC1	0010001	1	0110001	Q	1010001	q	1110001
DC2	0010010	2	0110010	R	1010010	r	1110010
DC3	0010011	3	0110011	S	1010011	s	1110011
DC4	0010100	4	0110100	T	1010100	t	1110100
NAK	0010101	5	0110101	U	1010101	u	1110101
SYN	0010110	6	0110110	V	1010110	v	1110110
ETB	0010111	7	0110111	W	1010111	w	1110111
CAN	0011000	8	0111000	X	1011000	x	1111000
EM	0011001	9	0111001	Y	1011001	y	1111001
SUB	0011010	:	0111010	Z	1011010	z	1111010
ESC	0011011	;	0111011	[	1011011	{	1111011
FS	0011100	<	0111100	\	1011100	|	1111100
GS	0011101	=	0111101	]	1011101	}	1111101
RS	0011110	>	0111110	^	1011110	~	1111110
US	0011111	?	0111111	_	1011111	DEL	1111111

Another way of showing the same information is shown in Table 4 below.

Binary (Hexadecimal)								Table 4
000 (0)	001 (1)	010 (2)	011 (3)	100 (4)	101 (5)	110 (6)	111 (7)	Bin (Hex)
NUL	DLE	Space	0	@	P	`	p	0000 (0)
SOH	DC1	!	1	A	Q	a	q	0001 (1)
STX	DC2	"	2	B	R	b	r	0010 (2)
ETX	DC3	#	3	C	S	c	s	0011 (3)
EOT	DC4	$	4	D	T	d	t	0100 (4)
ENQ	NAK	%	5	E	U	e	u	0101 (5)
ACK	SYN	&	6	F	V	f	v	0110 (6)
BEL	ETB	'	7	G	W	g	w	0111 (7)
BS	CAN	(	8	H	X	h	x	1000 (8)
HT	EM	)	9	I	Y	i	y	1001 (9)
LF	SUB	*	:	J	Z	j	z	1010 (A)
VT	ESC	+	;	K	[	k	{	1011 (B)
FF	FS	,	<	L	\	l	|	1100 (C)
CR	GS	-	=	M	]	m	}	1101 (D)
SO	RS	.	>	N	^	n	~	1110 (E)
SI	US	/	?	O	_	o	DEL	1111 (F)

There are a couple of important points that needs to be made about ASCII here. First, as already mentioned, ASCII is used to represent letters, numbers and punctuation marks. But ASCII is also used for *control characters*. Some control characters can perform simple actions and examples include: *carriage return* (CR), *line feed* (LF) and *form feed* (FF). Other control characters can assist in establishing and maintaining the data communications link and examples include: *acknowledged* (ACK), *not acknowledged* (NAK), *start of header* (SOH) and *null* (NUL).

Second, as ASCII codes look like 7-bit binary numbers, they are often converted to, and written in, hexadecimal form rather than using zeros and ones. For example, the ASCII code for upper case letter aitch (that is, H) is 1001000 but you're more likely to see it written as $48.

This can be confusing for the new-comer to digital electronics because $48 can be the hexadecimal equivalent of the ASCII code for the letter aitch or a hexadecimal number equivalent to the number 72_{10}. Only experience will help you correctly interpret information written in hexadecimal as either true numbers or ASCII characters.

Third, ASCII is a true code. This means that the columns are not weighted and so ASCII cannot be used for mathematical operations. This becomes immediately obvious if you add the ASCII codes for the numbers 7_{10} and 8_{10} (0110111 and 0111000 respectively). The result is 1101111 but this code represents the letter o in ASCII and not the number 15_{10}.

Converting alphanumeric or control characters to ASCII

The procedure to convert alphanumeric/control characters to BCD is described below with each step being demonstrated using the example of converting the letter *b* to ASCII.

1. Find the alphanumeric character in the table (use Table 4 rather than Table 3).

Binary (Hexadecimal)								
000 (0)	001 (1)	010 (2)	011 (3)	100 (4)	101 (5)	110 (6)	111 (7)	Bin (Hex)
NUL	DLE	Space	0	@	P	`	p	0000 (0)
SOH	DC1	!	1	A	Q	a	q	0001 (1)
STX	DC2	"	2	B	R	b	r	0010 (2)
ETX	DC3	#	3	C	S	c	s	0011 (3)
EOT	DC4	$	4	D	T	d	t	0100 (4)
ENQ	NAK	%	5	E	U	e	u	0101 (5)
ACK	SYN	&	6	F	V	f	v	0110 (6)

2. Follow the column to the top and note the binary or hexadecimal number.

Binary (Hexadecimal)								
000 (0)	001 (1)	010 (2)	011 (3)	100 (4)	101 (5)	110 (6)	111 (7)	Bin (Hex)
NUL	DLE	Space	0	@	P	`	p	0000 (0)
SOH	DC1	!	1	A	Q	a	q	0001 (1)
STX	DC2	"	2	B	R	b	r	0010 (2)
ETX	DC3	#	3	C	S	c	s	0011 (3)
EOT	DC4	$	4	D	T	d	t	0100 (4)
ENQ	NAK	%	5	E	U	e	u	0101 (5)
ACK	SYN	&	6	F	V	f	v	0110 (6)

3. Follow the row to the end and note the binary or hexadecimal number.

Binary (Hexadecimal)								
000 (0)	001 (1)	010 (2)	011 (3)	100 (4)	101 (5)	110 (6)	111 (7)	Bin (Hex)
NUL	DLE	Space	0	@	P	`	p	0000 (0)
SOH	DC1	!	1	A	Q	a	q	0001 (1)
STX	DC2	"	2	B	R	b	r	0010 (2)
ETX	DC3	#	3	C	S	c	s	0011 (3)
EOT	DC4	$	4	D	T	d	t	0100 (4)
ENQ	NAK	%	5	E	U	e	u	0101 (5)
ACK	SYN	&	6	F	V	f	v	0110 (6)

4. Put the two binary or hexadecimal numbers together to give the ASCII code for the character.

Binary (Hexadecimal)								
000 (0)	001 (1)	010 (2)	011 (3)	100 (4)	101 (5)	110 (6)	111 (7)	Bin (Hex)
NUL	DLE	Space	0	@	P	`	p	0000 (0)
SOH	DC1	!	1	A	Q	a	q	0001 (1)
STX	DC2	"	2	B	R	b	r	0010 (2)
ETX	DC3	#	3	C	S	c	s	0011 (3)
EOT	DC4	$	4	D	T	d	t	0100 (4)
ENQ	NAK	%	5	E	U	e		0101 (5)
ACK	SYN	&	6	F	V		v	0110 (6)

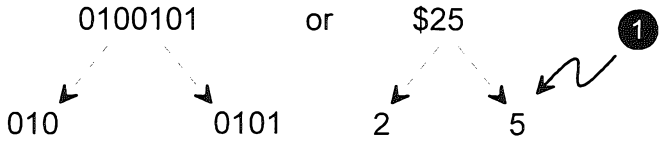
1100010 or $62

This example demonstrates that 1100010 or $62 is the ASCII code for the letter b.

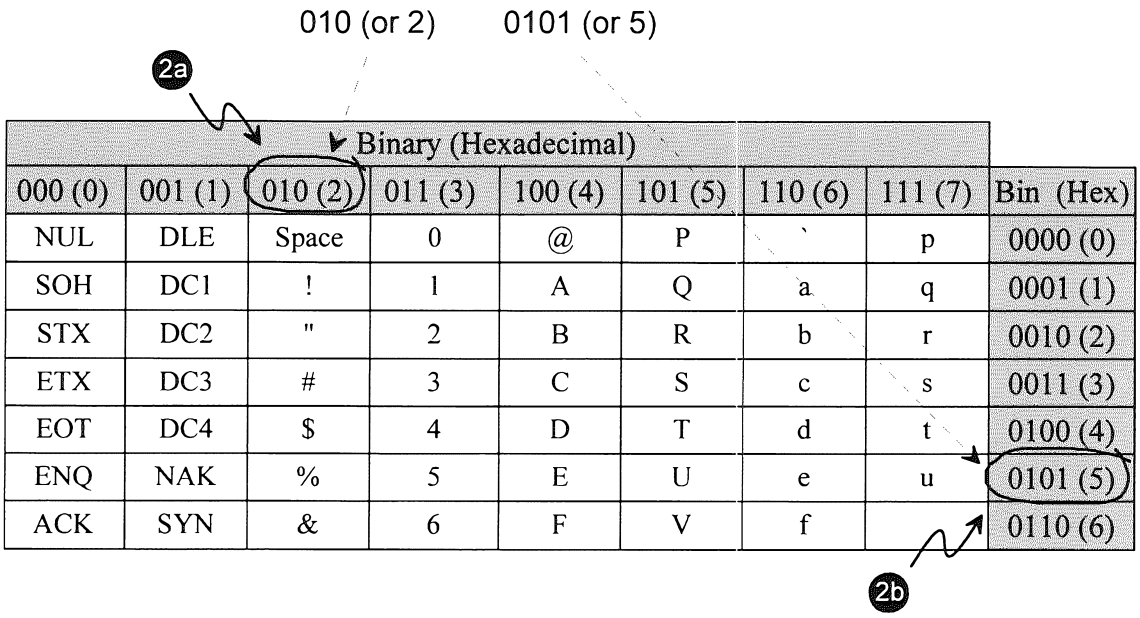
Converting ASCII codes to alphanumeric characters

The procedure to convert ASCII codes to alphanumeric/control characters is described below with each step being demonstrated using the example of converting the ASCII code $25 to its equivalent character.

1. If the ASCII code is written using zeros and ones then split it into a group of four bits and three bits starting from the right hand side. If it's written in hexadecimal then simply split the two digits.



2. If the ASCII code is written using zeros and ones then locate the 3-bit number across the top of the table and the 4-bit number across the side of the table. If it's written in hexadecimal then locate the left side digit across the top of the table and the right side digit across the side of the table.



		Binary (Hexadecimal)						
000 (0)	001 (1)	010 (2)	011 (3)	100 (4)	101 (5)	110 (6)	111 (7)	Bin (Hex)
NUL	DLE	Space	0	@	P	`	p	0000 (0)
SOH	DC1	!	1	A	Q	a	q	0001 (1)
STX	DC2	"	2	B	R	b	r	0010 (2)
ETX	DC3	#	3	C	S	c	s	0011 (3)
EOT	DC4	$	4	D	T	d	t	0100 (4)
ENQ	NAK	%	5	E	U	e	u	0101 (5)
ACK	SYN	&	6	F	V	f		0110 (6)

3. Follow the column down and the row across until they intersect. The character in that cell is the character represented by the ASCII code.

Binary (Hexadecimal)								
000 (0)	001 (1)	010 (2)	011 (3)	100 (4)	101 (5)	110 (6)	111 (7)	Bin (Hex)
NUL	DLE		0	@	P	`	p	0000 (0)
SOH	DC1	!	1	A	Q	a	q	0001 (1)
STX	DC2		2	B	R	b	r	0010 (2)
ETX	DC3		3	C	S	c	s	0011 (3)
EOT	DC4		4	D	T	d	t	0100 (4)
ENQ	NAK	%						0101 (5)
ACK	SYN	&	6	F	V	f	v	0110 (6)

This example demonstrates that the percentage sign (%) is the character represented by the ASCII code $25.

Practise converting alphanumeric/control characters to ASCII and vice versa for yourself by trying the following questions. Note: Use Table 4 (on page 7-7) to help with your conversions.

1. Determine the ASCII code (in both binary and hexadecimal) for the following characters:

   (a) SUB

   (b) ?

   (c) Y

   (d) {

2. Determine the character for the following ASCII codes:

   (a) $41

   (b) 0111001

   (c) $1D

   (d) 1101101

Applications for ASCII

ASCII was developed in the 1960s for late-model teleprinter or teletype machines like those used in the *Telex* service. Telex provided a type of direct printer-to-printer communications system that could operate over the global public telephone network to facilitate cheaper international communications that didn't need to occur in real time (eg request for quotations from overseas suppliers in different time-zones). This was in a time when international phone calls cost well in excess of $6/minute in 2024 dollars so the Telex service made good economic sense.

ASCII was subsequently adopted for use with early computers for uses including: the implementation of command line interfaces (CLIs) like DOS; for the creation and printing of basic text documents (eg .txt files) before wordprocessors were developed, and for communications between computers.

There are few, if any teletype systems around today but ASCII is still used for low-level hardware applications (again, for command line interfaces) and in legacy systems (ie old computer systems that are still in use today and so still need support).

That said, ASCII is still very widely used today indirectly because it forms the basis of *Unicode* (more below). The first 128 codes in Unicode are the same as the original 7-bit ASCII character code set.

Unicode

Prior to the development of Unicode, there were multiple character codes in use to render textual information. This included Morse code which dated all the way back to 1838. These character codes were contextual in that they only worked within the field of technological practice in which they originated and subsequently spawned or absorbed. Morse for code instance was developed for use in the cable telegraph system but then went on to be used in radiotelegraphy. Then as radiotelegraphy became increasingly used in the maritime and aviation sectors (for their communications and navigation) Morse code became highly associated with them too.

The various character codes were developed in isolation of each other and so they were completely incompatible. As a consequence, a message or file created using one character code would not render the same text or information in another character code. (To be fair, there was really no pressing need for compatibility between the different systems as they largely operated independently of each other. Moreover, where information interchange between systems was required, the volumes involved were readily satisfied by manual conversion.)

By the 1980s there was a realisation that a universal system of character encoding would be useful and efforts were begun to develop a system that culminated in Unicode (1.0), the standard for which was first published in 1991. The Unicode Consortium was established in the same year to maintain and update the standard as necessary.

Unicode's advent was certainly propitious because the worldwide web was invented in 1992 and, given its global nature, it was a technology that would benefit immensely from a universal system of encoding characters. Today, it's estimated that something like 98% of websites use Unicode to render text on computer screens.

Where the original ASCII could only support 128 characters, Unicode supports almost 150,000 characters which apparently is enough for the overwhelming majority of written characters of languages in wide use in the world today. The interesting thing about Unicode though is that, although it's based on ASCII, it's a much more clever system than simply increasing the number of bits in the code (to 21 bits today) to encompass all those characters.

To explain, when Unicode was first developed, the power of microprocessors, and the storage capacity of memory systems, couldn't accommodate a system with so many bits. (Note, we are talking about the early 1990s, right!) So, the Unicode Consortium devised an encoding system that allowed computers and other devices to use an n-bit code that would that could access a set of the characters for a script from the map of the entire character encoding using as few as eight bits (UTF-8). [Your author won't explain it in any more detail than that because he can't.]

Gray code

As you'll learn in a later chapter, digital electronics is used extensively to control things. When the device being controlled is something that rotates (like a motor), it's often necessary for the electronics to "know" the current position of the shaft or axle so the amount needed to turn it to a new position can be calculated. Devices called *rotary encoders* were designed for doing this and a widely used type is the *optical encoder*. These devices can be used for rotational input controls too, instead of a potentiometer. If you've ever adjusted a rotary volume control on a stereo and noticed that the knob can turn indefinitely in either direction, then this was probably an optical encoder.

Optical encoders employ a disk similar to (but different in an important way) the one shown in Figure 1 below. The disk would be mounted on an axle or rotary shaft and read optically by shining light (typically infrared) onto the disk and using sensors to detect whether any of the light that is reflected. Reflected light from the white areas of the disk are read as logic-0 and the absence of reflected light due to the dark areas of the disk are read as logic-1 (or it can be the other way around).

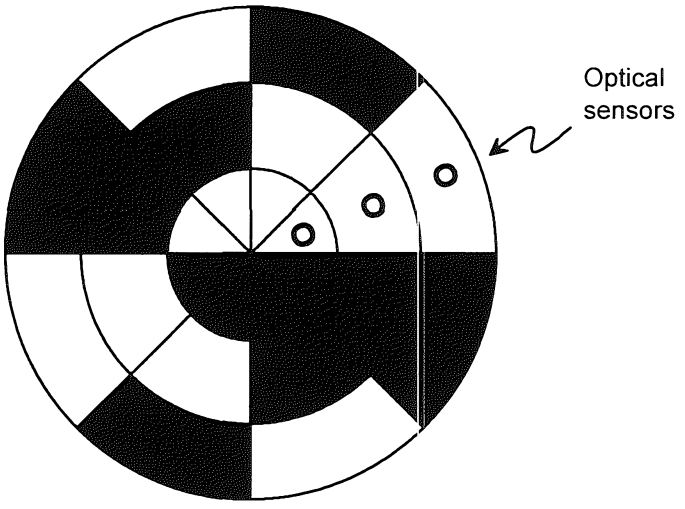


Figure 1 *A binary optical encoding disk (not used - read on to see why)*

As the shaft rotates clockwise, the photoelectronics (one sensor for each bit in the code) read the disk sending the information to the controlling digital electronics. The disk in Figure 1 has eight codes (from 000 to 111) which means that the resolution of the shaft's rotary position is only 45°. That is, the disk must turn 45° before the code changes. For a better resolution more bits are necessary but of course this requires more optical sensors so there's a trade-off.

Importantly, although the basic idea of rotary position sensing has been explained here using a binary encoded disk, binary cannot be used for this purpose. The reason for this is two-fold. First, low-cost optical sensing is not very precise. That is, although the difference between the logic levels on the disk is sharp, the exact point where the change is *detected* will not necessarily be the same every time.

Second, there are several instances where two or more bits change for a single change in the binary number. For example, the transition from 000 to 001 requires only one bit to change. However, the transition from 001 to 010, 011 to 100 and 101 to 110 all involve two bits changing. Worse, the transition from 111 to 000 involves three bits changing.

These two issues combine to generate invalid codes between binary number transitions on the disk. To explain, when the binary code on the disk changes from 001 to 010 the change in the two bits are unlikely to be read at the same time. This means that it's likely for the optics to read the number 011 or 000 very briefly on the transition. In other words, the digital electronics can be sent the following information: 001⇨**011**⇨010 or 001⇨**000**⇨010.

These brief transitory codes risk confusing the electronics regarding the shaft's instantaneous rotary position. There are two solutions to this problem. One is to improve the precision of the optics by using lasers but this will also increases the cost. The second more affordable approach is to use Gray code on the disk instead of binary.

Gray code is one of several classed as *minimum change* codes in which only one bit changes for each successive code in the sequence. If you look carefully, you can see this in the 3-bit Gray code in Table 1 below.

Table 1

Decimal	Binary	Gray code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Copyright: Not to be reproduced without permission

A 3-bit Gray disk is shown in Figure 2 below.

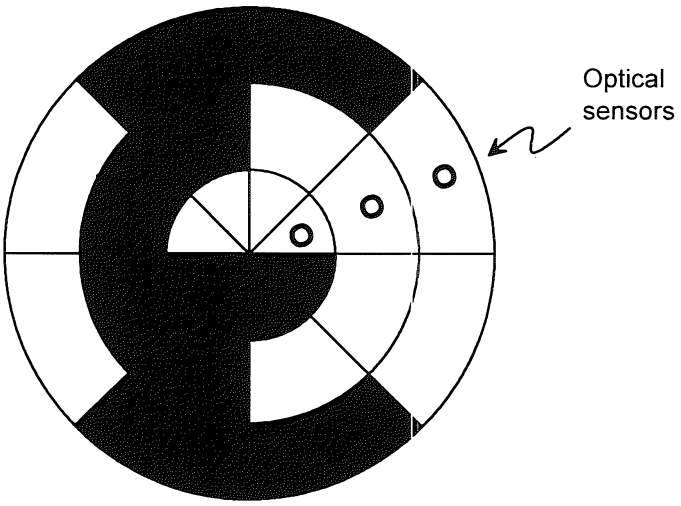


Figure 2 *A Gray code optical encoding disk*

Gray code solves the problem of brief transitory codes because, with only one bit changing each time, transitory codes are impossible. The disadvantage is that the rotary encoder generates a non-binary code. However, this is only problematic where mathematical operations must be performed. In these situations, the Gray code must be converted to binary (or BCD). [**Note:** It's possible to manually convert between Gray code and binary but you'll not be asked to do so for the assessments.]

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. Convert the following decimal numbers into BCD.

   (a) 7

   (b) 10

   (c) 29

   (d) 358

   (e) 1,476

2. Convert the following BCD numbers to decimal.

   (a) 0011

   (b) 00100001

   (c) 10011000

   (d) 010001010111

   (e) 0001001001010011

3. Explain why the number 00011011 must be a binary number and not a BCD number.

4. What is an advantage of BCD over binary?

 Copyright: Not to be reproduced without permission

5. What is a disadvantage of BCD compared to binary?

6. Name two pieces of electronics equipment in which you would find BCD being used.

7. Convert the following alphanumeric/control characters to ASCII. Show the ASCII codes both in binary and hexadecimal.

   (a) 1

   (b) u

   (c) ETB

   (d) >

   (e) ~

8. Convert the following ASCII codes to alphanumeric/control characters.

   (a) $06

   (b) 0111010

   (c) $1B

   (d) 1111111

   (e) 0111101

9. Explain why the number 0101000 must be an ASCII code and not a BCD number.

10. Name a common application for *Unicode*.

11. What's the difference between binary and Gray code?

12. Why is this an important difference for optical rotary position encoding of motors etc?

13. What two transitory codes can be generated when a binary encoded optical disk changes from the number 101 to 110?

Copyright: Not to be reproduced without permission

Student notes

Section 8 Digital displays

Purpose To develop an understanding of the operation of display technologies commonly used in digital electronics equipment.

Objectives At the end of this section you should be able to:

- Draw the schematic symbol for light emitting diodes (LEDs) and identify the anode and cathode pins
- Explain the purpose of the series resistor in LED circuits
- Calculate the resistance of a suitable series resistor in LED circuits given the operating parameters for the LED and the EMF voltage
- Explain the operation of the circuit that would operate an LED from a logic gate
- Explain why an LED can be current sunk and not current sourced by a logic gate
- List equipment that use LED seven-segment displays
- Label the segments in a seven-segment display
- State the condition of each segment in a seven-segment display when displaying the numbers 0 to 9
- Explain the difference between common anode and common cathode LED seven-segment displays
- Explain the function of display decoder/driver ICs
- List the advantages and disadvantages of LED seven-segment displays
- List equipment that use liquid crystal displays (LCDs)
- Explain the basic operation of reflective-mode and transmissive mode LCDs
- State the electrical requirements to turn on an LCD segment
- Explain the operation of the circuit that would operate a segment in an LED from an XOR gate
- List the advantages and disadvantages of LCDs
- Use a DMM and logic probe to test and measure the operation of digital displays

Introduction

It is usual for most digital electronics equipment to display information. The information displayed can be the simple indication of the equipment's mode of operation (for example, whether it is on or not), it can be numeric information (for example, the time on a digital clock) or it can be alphanumeric information (for example, a text message on a mobile phone or pager). In this section we'll learn about three display devices used in digital electronics: *light-emitting diodes*, *seven-segment displays* and *liquid crystal displays*.

Light emitting diodes

Light-emitting diodes (or LEDs) are a popular way of indicating simple information in digital electronics equipment. They are preferred over lamps (or bulbs) because of their supply requirements and efficiency.

Schematic symbols used to represent LEDs in circuit diagrams are shown below in Figure 1 below.

Figure 1 *Schematic symbols for light emitting diodes*

Like all diodes, LEDs are *polarised.* That is, one leg of the LED is positive (called the *anode*) and the other leg negative (called the *cathode*). The anode and cathode legs of LEDs are shown using the schematic symbol in Figure 2 below.

Figure 2 *The polarity of the leads on LEDs*

There are two ways that the anode and cathode legs can be identified on the LED itself. First, the legs of most LEDs are different lengths. Usually, the longer leg is the anode and the other leg is the cathode. This is shown in Figure 3 below.

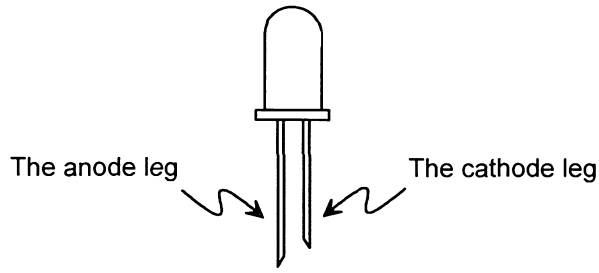


Figure 3 *One method of identifying the anode leg on LEDs*

Alternatively, if you look closely at an LED from underneath, you'll notice that one side of its base is probably chamfered. This is usual and the leg closest to the chamfered side is the cathode. This is shown in Figure 4 below.

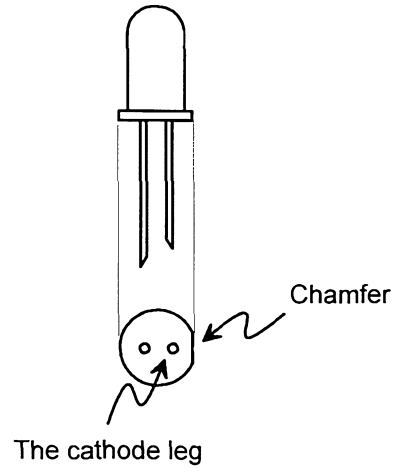


Figure 4 *The other method of identifying the anode leg on LEDs*

Operating an LED in a basic circuit

Figure 5 below shows the simplest circuit for making LEDs light up.

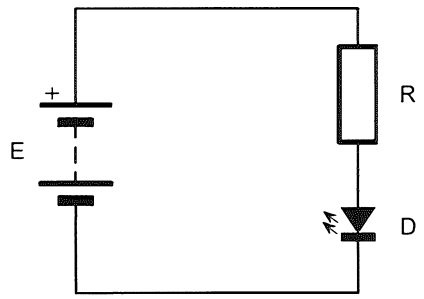


Figure 5 *The basic circuit that will light up LEDs*

Provided the EMF's voltage is big enough and the resistor is an appropriate value (issues we'll discuss next), current will flow around the circuit in the direction shown in Figure 6 below. The current through the resistor will develop a potential difference across the resistor and the diode in the polarity shown.

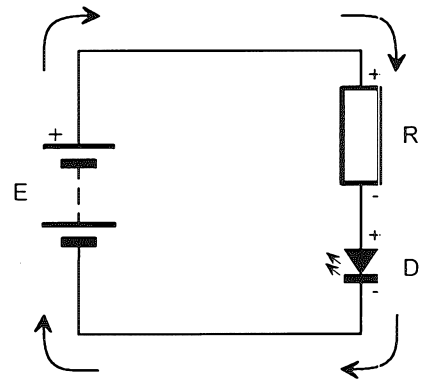


Figure 6 *The direction of conventional current in the circuit of Figure 5 and the polarity of the potential differences across the resistor and diode*

Importantly, the fact that LEDs are polarised means that, to operate them correctly, they must be connected in the circuit in the correct polarity. The anode must be made positive with respect to the cathode and this is the case in the circuit in Figure 6 above. If the LED is connected in the wrong polarity, current will not flow in the circuit and the LED wont light up. The LED may even be damaged by *reverse bias* voltages as low as 20V and less.

Typical operating parameters of LEDs

The resistance of the resistor in Figure 5 cannot just be any value. If it's too large, the LED will not turn on. But, if it's too small, excessive current could damage the LED. To turn on the LED and avoid excessive current, the resistor's resistance must be chosen carefully with attention the LED's operating parameters, the size of the EMF and Ohm's Law (naturally!).

The operating parameters of LEDs varies depending on colour but red LEDs typically require a forward voltage of 1.8V to 2.2V. The actual size of the forward voltage is a function of the LED's forward current (so, typically, 1.8V at 7mA, 2V at 20mA and 2.2V at 35mA). The LED's brightness is also a function of the forward current (so, typically, 1lm at 10mA, 2.5lm at 20mA and 4lm at 30mA).

The resistor's value is chosen so that it sets the circuit current to the desired operating current for the LED. For any given operating current, the resulting forward voltage drop will not vary much. This fact is demonstrated below in Figures 7 and 8 using LEDs with a forward current that gives a forward voltage of 1.8V. Figure 7 shows an EMF of 5V and Figure 8 shows an EMF of 10V. In both circuits the potential difference across the LED is 1.8V. As a consequence, the potential difference across the resistors are different (3.2V and 8.2V respectively).

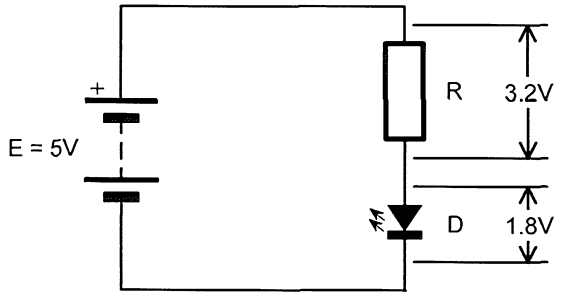


Figure 7 With an EMF of 5V and an LED operating voltage of 1.8V, the potential difference across the resistor is 3.2V

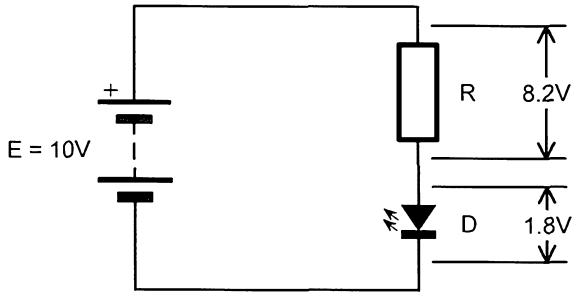


Figure 8 With an EMF of 10V and an LED operating voltage of 1.8V, the potential difference across the resistor is 8.2V

That being the case, the resistor's value is determined using Ohm's Law, the potential difference across itself and the required current for the LED. The equation that can be used to calculate the resistor's value in this circuit is:

$$R = \frac{V_R}{I_{LED}}$$

Where:
V_R is the potential difference across the resistor
I_{LED} is the LED's operating current

In the circuit of Figure 7, if the LED's nominal operating current is 7mA, the resistor's value should be:

$$R = \frac{V_R}{I_{LED}}$$

$$R = \frac{5V - 1.8V}{7mA}$$

$$R = 457\Omega$$

Although 457Ω is not a preferred resistor value (in the E12 system), 390Ω and 470Ω are so either of these can be used.

If a 390Ω resistor is used, the forward current would be a little bigger and so too would the forward voltage. Conversely, if a 470Ω resistor is used, the forward current would be a little smaller and so too would the forward voltage. There would be differences too in the LED's brightness but it would be too small for the human eye to perceive.

Practise determining the value of a LED's series resistor by trying the following questions.

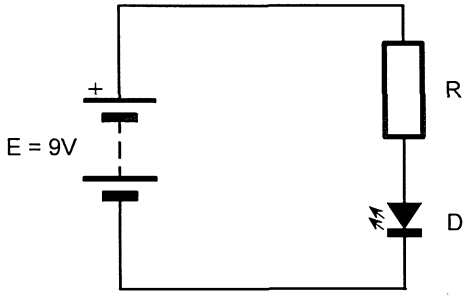


Figure 9

1. What should the resistor's value be in Figure 9 if the blue LED has an operating voltage of 3.4V and a nominal operating current of 20mA?

2. What preferred resistor value could be used in the circuit?

Driving LEDs from logic devices

In digital electronics circuits it is common to use a logic device like a logic gate to turn LEDs on and off. A circuit that allows a logic gate to control LEDs is shown below in Figure 10 using an AND gate (though any of the gates could have been used). The resistor's value is calculated in the same way as for the basic circuit discussed previously.

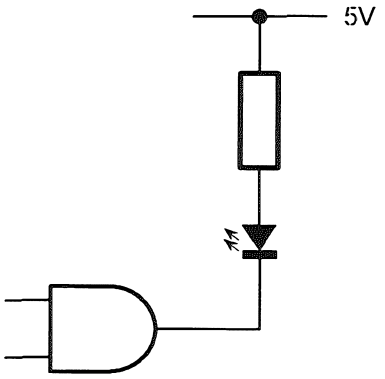


Figure 10 *The circuit of an AND gate used to turn on/off LEDs*

Operation

When the gate's output is a logic-0, the LED is on. To explain why, recall that a logic-0 is electrically represented in digital electronics circuits by 0V. So, with the gate's output at 0V, there is a 5V potential difference across the series LED and resistor combination. This means that current can flow through the resistor and LED turning the LED on. This is shown in Figure 11 below.

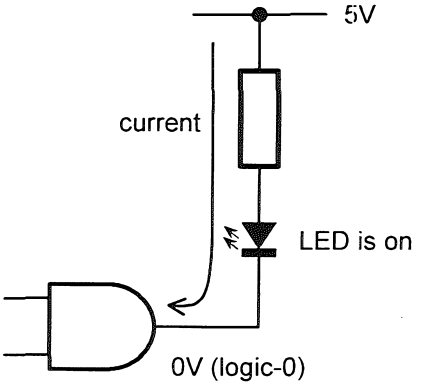


Figure 11 *When the gate's output is a logic-0 the LED is on*

When the gate's output is a logic-1, the LED is off. To explain why, recall that a logic-1 is electrically represented in digital electronics circuits by 5V. So, with the gate's output at 5V, there is a 0V potential difference across the series LED and resistor combination. This means that no current can flow through the resistor and LED and so the LED is off. This is shown in Figure 12 below.

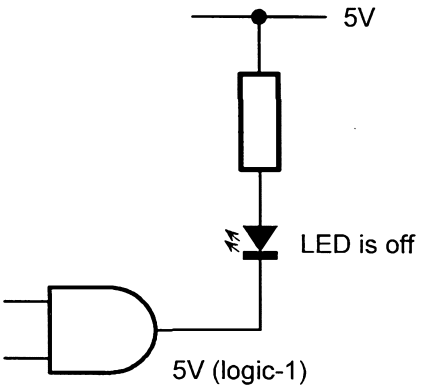


Figure 12 *When the gate's output is a logic-1 the LED is off*

Importantly, the logic device that is used to control the LED must be able to sink its operating current. Put another way, the logic device's I_{OL} must be greater than the LED's operating current.

If you look at the data in Table 4 on page 7-11 you'll see that most of the 74 series TTL logic devices can comfortably sink the minimum operating current of **typical** LEDs. However, the typical I_{OL} of CMOS devices is only 1mA (see Table 8 on page 7-19) and so CMOS devices cannot sink the current of most LEDs.

Basic theory would suggest that the circuit in Figure 13 below can also operate an LED. A logic-0 (which is 0V) should theoretically turn the LED off and a logic-1 (which is 5V) should turn it on.

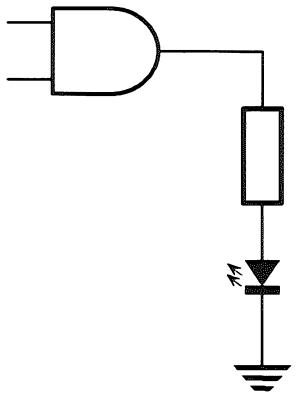


Figure 13 *Current sourcing an LED from a logic device like this usually doesn't work*

However, if you refer to Table 4 on page 7-11 again you'll see that the maximum source current for any of the 74 series TTL logic gates is 1mA (the others are well below this at 0.4mA). This is below the minimum operating current of most LEDs and so this circuit will not work effectively.

Seven-segment displays

Although LEDs are very useful, they have obvious limitations as display devices when used individually. A single LED can't display much information. An improvement is the seven-segment display.

As you almost certainly already know, seven-segment displays use seven segments arranged in a figure-eight pattern to make decimal numbers from zero to nine by turning on or off the required segments. This arrangement is shown below in Figure 14. Notice that the segments are labelled using lower case letters from *a* to *g* starting from the top and this is standard.

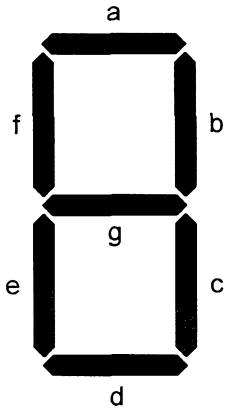


Figure 14 *The arrangement of the segments in a seven-segment display*

Figure 15 below shows how the digits 0 to 9 can be made using two or more of the segments.

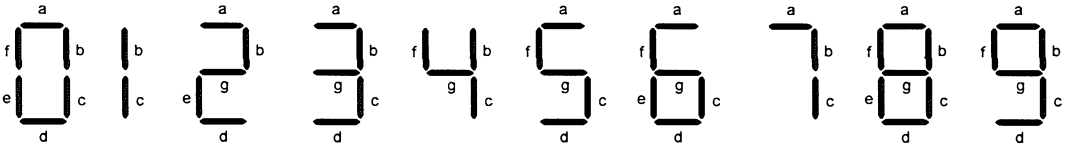


Figure 15 *The numbers 0 to 9 displayed using a seven-segment display*

Seven-segment displays can be implemented using a range of technologies. A common example involves the use of LEDs for the segments. Other technologies include liquid crystal displays (such as the display on your calculator) and vacuum fluorescent displays (such as the display on a typical PVR).

LED seven-segment displays

Seven-segment displays made using LEDs were once very popular but are becoming less so these days because of new technologies. Having said that, they're still commonly used for the digital clock on clock-radios, the clock/timer display on electric ovens and the readout on test equipment such as power supplies, function generators and frequency counters.

The LED seven-segment display package is usually rectangular and contains eight LEDs not seven. The extra LED is for a decimal point.

Having eight LEDs suggests that the package should have sixteen pins on it (two for each LED). However, most packages have fewer than sixteen pins because either the anode or the cathode leg of all eight LEDs are connected together inside the package. This means that seven fewer pins are needed on the package and this is done to reduce external wiring.

When all of the LEDs' anode legs are connected together inside the display, the device is said to be a *common anode* display. When all of the LEDs' cathode legs are connected together the device is said to be a *common cathode* display. Figures 16a and 16b below show the internal wiring of common anode and common cathode LED seven-segment displays (not including the LED for the decimal point).

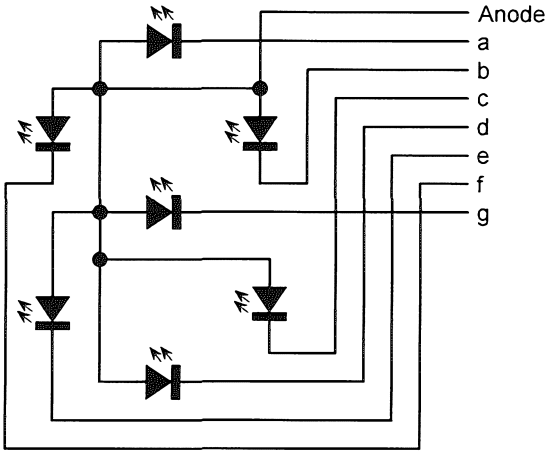


Figure 16a *Common anode display*

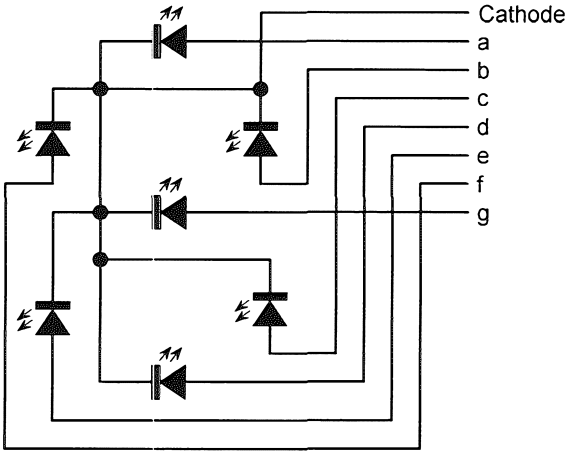


Figure 16b *Common cathode display*

To turn on a segment in a common anode display, the display's anode pin must be connected to some positive voltage (like 5V) and the segment's pin must be asserted with a logic-0 (via a resistor to limit current). To turn on a segment in a common cathode display, the display's cathode pin must be connected to ground (or common) and the segment's pin must be asserted with a logic-1 (via a resistor to limit current).

Seven-segment display decoder/driver chips

As already mentioned, digital electronics circuits turn on and off the appropriate LEDs in the seven-segment display to make the required number. This must be managed in some way and a circuit like the circuit in Figure 10 (page 8-8) could be used. However, there are two practical problems with doing so.

First, each LED in the seven-segment display needs its own logic gate to control it so a total of seven logic gates are needed (one for each segment). As there are typically only four 2-input gates per digital IC, two ICs would be needed to control the entire digit. Moreover, if the display consisted of four digits (like on a digital clock) then a total of eight digital ICs would be needed.

Second, additional *decoding* circuitry (and hence more ICs) is needed to convert the binary or BCD numbers that the digital system would use into a 7-bit code that could turn on and off the appropriate LEDs. To explain, the 4-bit binary and BCD number for zero is 0000 but there's not enough digits in 0000 to control the seven-segments.

These two problems combine to make the circuitry needed to operate a seven-segment display require quite a number of ICs. Having to use so many ICs would be expensive and would require a physically large printed circuit board (PCB) to accommodate them all. To overcome these problems, manufacturers of ICs have developed chips with all the circuitry needed to *decode* binary or BCD numbers and *drive* the seven LEDs built in. These chips are known as *decoder/driver* chips.

Readily available decoder/driver ICs include the 7446 and 7447 which drive common anode seven-segment displays and the 7448 and 7449 which drive common cathode seven-segment displays. All four chips require a BCD input and produce a number on the display that is the decimal equivalent of the BCD input.

Figure 17 below shows the circuit of a 7447 BCD to seven-segment decoder/driver IC driving a common anode LED seven-segment display.

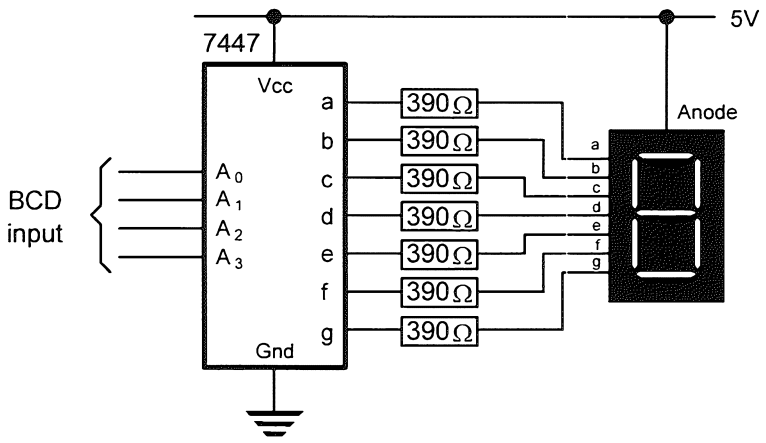


Figure 17 *The 7447 driving a common anode LED seven-segment display*

The resistors' values are calculated in the same way as for the circuits discussed earlier.

The seven-segment display arrangement (whether implemented using LEDs or LCD technology) can show 128 unique patterns. However, over half of these are meaningless for everyday communication in English and most other western languages too. The numbers 0 to 9 can be shown, as well as a handful of upper or lower-case letters (like A, H, U, t and u) but that's about it. For this reason, displays using fourteen and sixteen segments are available which are capable of showing vastly more unique patterns permitting not just numbers to be displayed but the whole alphabet either in upper-case (14-segments) or in both upper and lower-case (16-segments). A large number of symbols recognisable to the western writing tradition can also be implemented.

That said, displays using segments are a much more limited compared to dot matrix displays. LED dot matrix displays are available and you have possibly seen them used for the electronic noticeboard on Tangara and first generation (or A-set) Waratah trains on the Sydney rail network.

Advantages and disadvantages of LED displays

LED seven-segment displays can be seen in the dark whereas liquid crystal displays (LCDs) cannot (at least, not without a backlight). So, they are preferred in applications where it is expected that the display will be viewed at the night or other low light situations (hence why they are used for the display of most bedside digital clocks). However, LEDs require more current than LCDs to operate so they consume more power which makes them unsuitable for battery operated devices. LED dot matrix displays are difficult to implement on a small-scale. Also, LED displays are not very easy to read in strong back-light situations (that is, where there is a strong light behind the display). You may have noticed this problem when trying to read the time in the day-time on a digital bedside clock that is near a window.

Liquid crystal displays

Liquid crystal displays overcome some of the disadvantages of LED seven-segment displays and so have become very widely used. There are two basic types of liquid crystal display: reflective-mode and transmissive-mode.

Reflective-mode LCDs

The appearance and physical construction of reflective-mode LCDs is shown in Figure 18 below. This device would display several alphanumeric digits and not just one (on a digital multimeter for example).

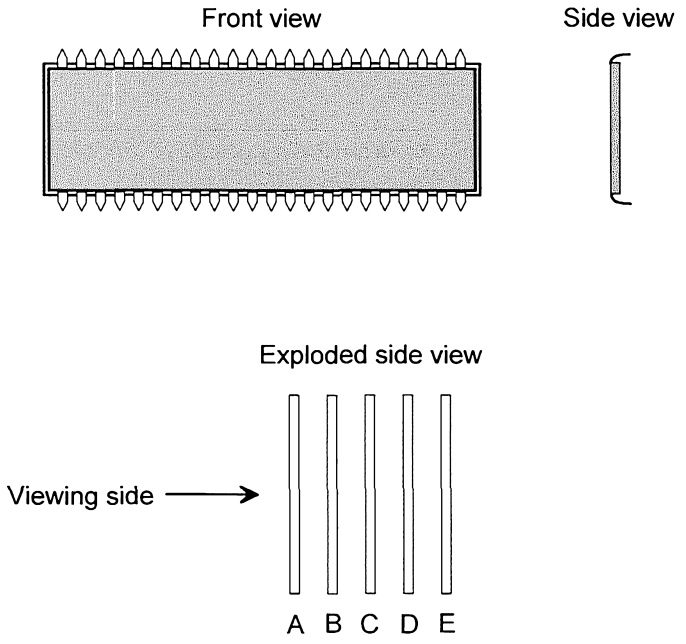


Figure 18 *The appearance and physical construction of reflective-mode LCDs*

Layer A is a length of polarised glass. Layer B is a length of non-polarised glass with transparent electrodes in the shape of segments (and other symbols) bonded to it. These clear electrodes are faintly visible if you hold the LCD at certain angles to the light. Layer C is a layer of liquid crystal in which the crystals are randomly aligned. Layer D is a another length of non-polarised glass with one large transparent electrode bonded to it. This electrode is usually referred to as the *backplane electrode*. Finally, layer E is a mirror.

Operation

The operation of a reflective-mode LCD relies on incident light (that is, light from the sun or electric lighting inside a room) entering the display from the viewing side, travelling through layers A, B, C and D and being reflected back through the layers and out of the viewing side by the mirror (layer E). This is shown in Figure 19 below.

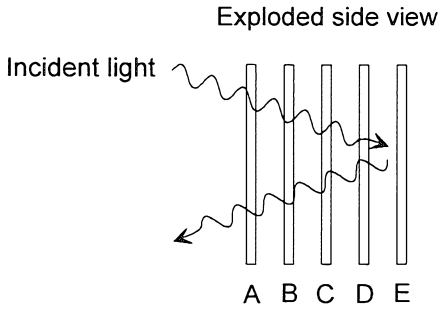


Figure 19 *Incident light enters the display from the viewing side and is reflected back to the viewer's eyes*

Figure 20 below shows the light entering and exiting the display from the viewing side on a three-dimensional drawing.

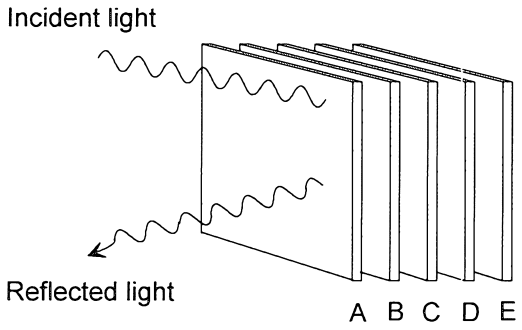


Figure 20 *A three-dimensional view of Figure 19*

Note that, although Figures 19 and 20 show only one light-wave at one angle, in reality there are an infinite number of light-waves entering and exiting the display from all sides across the length of the screen at an infinite number of angles. Importantly, this is happening whenever the LCD is exposed to light even if the display is not being operated. And, it is the light travelling through the polarised layer and the liquid crystal that gives the characteristic greyish hue that reflective-mode LCDs usually have.

This process of light travelling through the layers of the display and being reflected back to the viewer's eyes occurs when the display is being operated but a segment (or pattern) is blanked out. The electrode for that segment is simply left turned off and so it remains invisible to the viewer (see Figure 21).

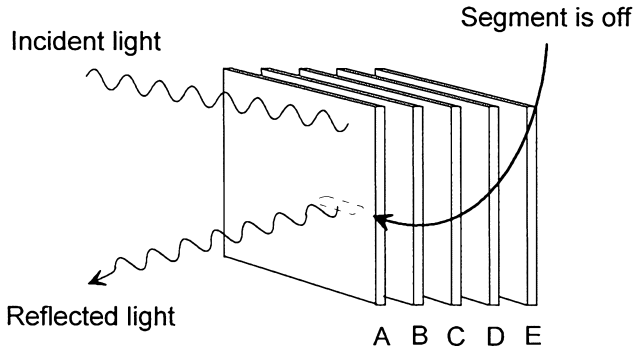


Figure 21 *A segment is off so remains invisible to the viewer*

When a segment is on, the electrode for that segment is activated. This causes the crystals in the layer of liquid crystal in the area between the segment's electrode and the backplane electrode to align in a direction 90° to the polarisation of light waves by layer A. This blocks light underneath the segment's electrode from being reflected back to the viewer's eyes (see Figure 22).

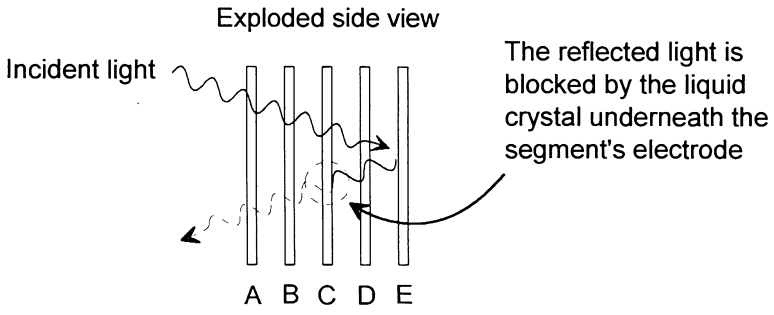

Exploded side view
Incident light
The reflected light is blocked by the liquid crystal underneath the segment's electrode
A B C D E

Figure 22 *The reflected light is blocked by the liquid crystal*

When this happens, light in the region of the electrode doesn't reach the viewer's eyes and so the screen turns black in the same shape as the electrode. If the electrode is shaped like a segment in a seven-segment display then the display would look like Figure 23 below.

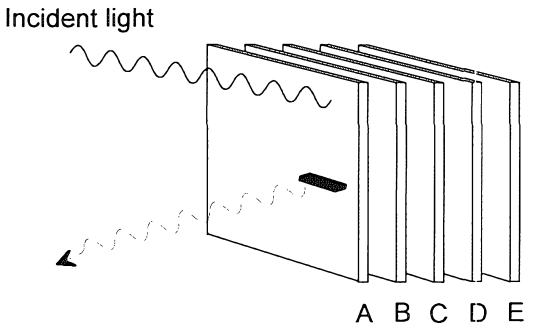


Figure 23 *The reflected incident light is blocked and so the display turns black in the same shape as the electrode*

Transmissive-mode LCDs

Transmissive-mode LCDs operate in the same basic way as reflective-mode LCDs but with one difference. Transmissive-mode LCDs don't have a mirror at the back of the display. So, instead of relying on light entering the display from the viewing side and being reflected back towards the viewers to be blocked or not blocked (depending on the segments and whether or not they're on) transmissive LCDs rely on light entering the display from behind. This is shown in Figure 24 below.

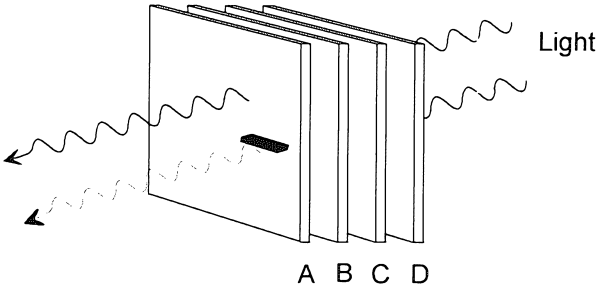


Figure 24 *The transmissive-mode LCD relies on light entering the display from behind*

Either incidental light or an electrical light source mounted behind the display can be used for transmissive-mode LCDs. More detailed information about LCDs can be found by visiting http://electronics.howstuffworks.com on the internet.

Turning the segments on and off

To turn a segment black, an AC potential difference must be developed between the segment's electrode and the backplane electrode. The method that is commonly used to do this involves connecting anti-phase squarewaves to each electrode and this is shown on the diagram in Figure 25 below.

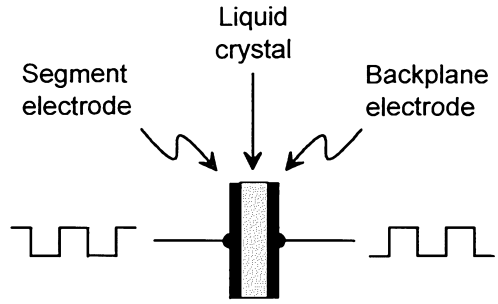


Figure 25 *Anti-phase squarewaves are connected to the two electrodes to turn the segment on and make it black on the screen*

As the squarewaves are in opposite phases, when the segment electrode is 5V the backplane electrode is 0V. And, when the segment electrode is 0V the backplane electrode is 5V. The resulting potential difference between the two electrodes squarewave.

To turn the segment off, one of the two squarewaves simply needs to be phase inverted so that both squarewaves have the same phase relationship (see Figure 26).

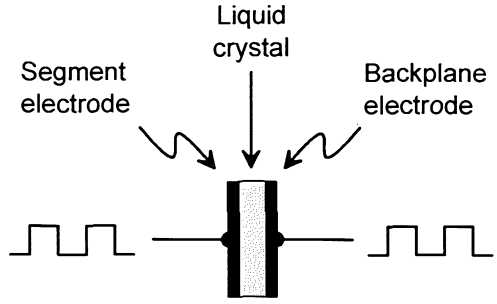


Figure 26 *In-phase squarewaves are connected to the two electrodes to turn the segment off and make it invisible on the screen*

When the segment electrode is 5V the backplane electrode is also 5V and the potential difference is 0V. Then, when the segment electrode is 0V the backplane electrode is also 0V and again the potential difference is 0V.

Figure 27 below shows a simple circuit that can be used to control individual segments to turn them on and off.

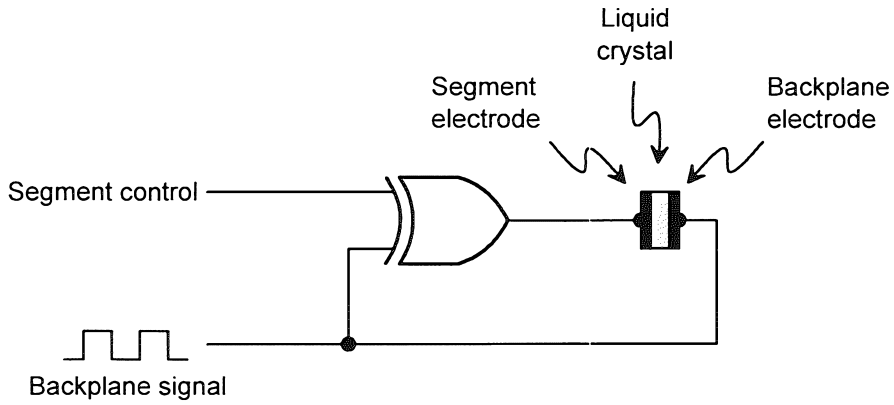


Figure 27 *A circuit that can be used to turn individual segments on and off*

As you can see, the backplane electrode is continuously supplied with a 5Vp-p squarewave, the frequency of which is typically between 30Hz and 60Hz. At the same time, the backplane signal is connected to one input of an XOR gate. The XOR gate's output is also a squarewave that is either in-phase with the backplane signal or anti-phase with it depending on the logic level on the segment control input.

When the segment control input is a logic-0 the XOR gate's output is either: a) logic-0 when the backplane signal is logic-0; or b) logic-1 when the backplane signal is logic-1. So the squarewave on the XOR gate's output is in the same phase as the backplane signal which means the segment is turned off and is invisible (see Figure 28).

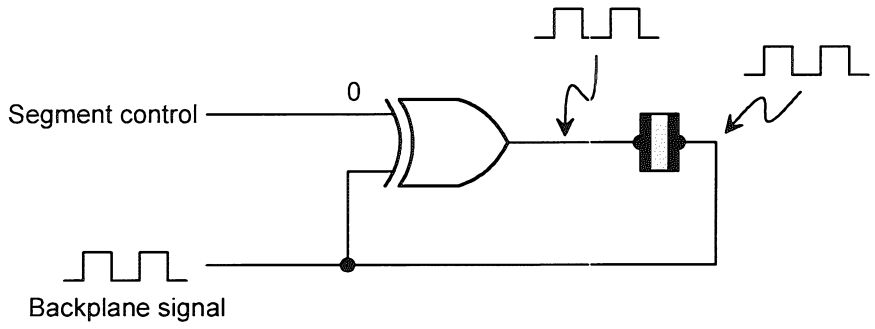


Figure 28 *When the segment control input is logic-0 the segment is turned off*

When the segment control input is a logic-1 the XOR gate's output is either: a) logic-1 when the backplane signal is logic-0; or b) logic-0 when the backplane signal is logic-1. So the squarewave on the XOR gate's output is in the opposite phase to the backplane signal which means the segment is turned on and is visible (see Figure 29).

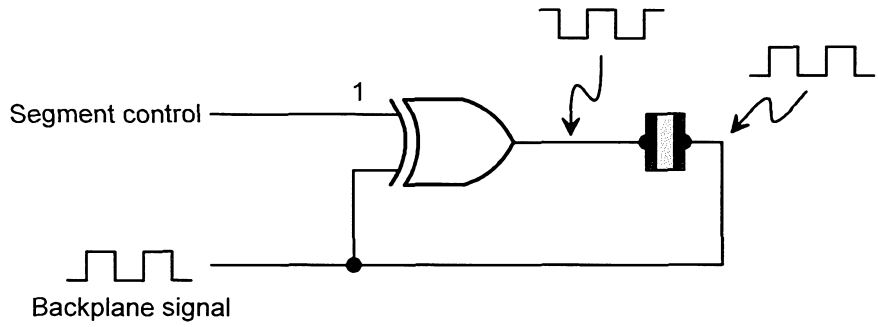


Figure 29 *When the segment control input is logic-1 the segment is turned on*

LCD decoder/driver chips

As already mentioned, the XOR control circuit shown above can be used to control only one individual segment (or pattern). This is important point because, although LCDs are a completely different technology to LED displays, they still display numbers using seven segments. So, controlling all of the segments in just one seven-segment digit would require seven of these XOR control circuits.

That being the case, LCDs have the same problem as LED seven-segment displays in terms of hardware requirements. It is for this reason that manufacturers design and sell decoder/driver chips for LCD displays as well as LED seven-segment displays.

Advantages and disadvantages of liquid crystal displays

LCDs require very small amounts of current to operate (less than 50μA per segment) and so they're much more efficient compared to LED displays making them ideal for battery operated devices (like calculators and digital multimeters). LCD dot matrix displays can be readily implemented at very small scale and your Casio calculator uses them. But LCDs can easily be designed to display a variety of symbols and shapes without the need for a dot matrix and you can also see example of these on your calculator if you look across the top of the display above the dot matrix arrangement.

The main disadvantage of LCDs is that the viewing angle is much more narrow that most other displays. So, if you move too far to the side of the display, LCDs become unreadable. Another disadvantage of LCDs is they can't be seen in the dark. This is an obvious limitation of their usefulness. Back-lighting helps but, when it is used, the display's contrast deteriorates and the power consumption rises dramatically potentially negating the efficiency advantage of LCDs.

Copyright: Not to be reproduced without permission

Student notes

Skill practice 8

An introduction to testing and measuring digital displays

This exercise is a practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 1½ hours to complete this exercise.

Equipment

- Digital trainer
- two DMMs
- logic probe
- 74LS00
- seven 390Ω ¼W resistors
- 470Ω ¼W resistor
- 560Ω ¼W resistor
- Red LED
- LSD5365 (7-segment display)
- 7447 IC
- banana leads
- hook-up wire
- wire strippers
- IC extractor

Remember:

Follow TAFE NSW WHS guidance at all times!

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

Part A - The LED

2. Gather the equipment needed for this exercise.

3. Read the following points and make sure that you perform them when wiring all circuits:

   - Handle all ICs carefully - remember ESD can damage them.
   - When plugging hook-up wire into the breadboard to make connections to the pins of the IC, use holes that are furthest away from the IC's pin.

4. Wire the circuit of Figure 1.

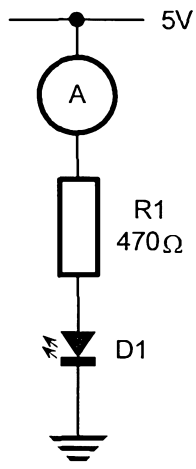


Figure 1

5. Turn on the power supply.

6. Measure the potential difference across the LED and record this in Table 1 (on the next page).

7. Measure and record the potential difference across the resistor.

8. Measure and record the circuit current.

9. Note and record the LED's brightness.

10. Turn off the digital trainer and swap the 470Ω resistor for a 390Ω resistor.

11. Repeat steps 5 to 9.

12. Turn off the digital trainer and swap the 390Ω resistor for a 560Ω resistor.

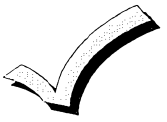
13. Repeat steps 5 to 9.

Table 1	R1 = 470Ω	R1 = 390Ω	R1 = 560Ω
Potential difference across the LED			
Potential difference across the resistor			
Circuit current			
LED's brightness			

Question 1
What is the relationship between the resistor's value and the circuit current?

Question 2
Why is the potential difference across the resistor the same regardless of the size of the circuit current?

Question 3
Was there much difference in the LED's brightness between the three resistor values?



The teacher needs to check your work at this point...

14. Wire the circuit of Figure 2.

    Note: Pinout data for the IC is at the back of the book.

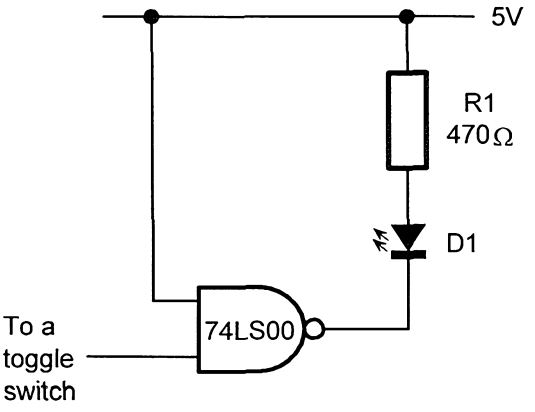


Figure 2

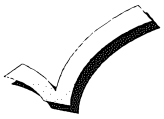
15. Turn on the digital trainer.
16. Set the toggle switch to logic-0.
17. Measure the logic level on the gate's output and record this in Table 2 (on the next page).
18. Note whether the LED is on or off and record your observation in Table 2.
19. Set the toggle switch to logic-1.
20. Repeat steps 17 and 18.

Copyright: Not to be reproduced without permission

Table 2	Output of gate	LED on or off?
Toggle switch set to logic-0		
Toggle switch set to logic-1		

Question 4
Explain why setting the toggle switch to logic-0 turns off the LED. Use the NAND gate's truth table to explain the first half of your answer.

Question 5
Explain why setting the toggle switch to logic-1 turns on the LED. Again, use the NAND gate's truth table to explain the first half of your answer.



The teacher needs to check your work at this point...

21. Set the *Frequency* control of the trainer's function generator to the "1" setting.

22. Set the *Range* control of the trainer's function generator to the *×1* position.

23. Disconnect the wire to the toggle switch and connect it to the function generator's ***TTL Mode*** output instead.

24. Watch what happens to the LED and record your observation below:

_____

25. Slowly rotate the *Frequency* control of the trainer's function generator to maximum. Watch what happens and record your observation below:

_____

26. Set the *Range* control of the trainer's function generator to the *×10* position. Watch what happens and record your observation below:

_____

Question 6
Why does the LED flash when the input to the NAND gate is connected to the function generator's *TTL Mode* output?

_____

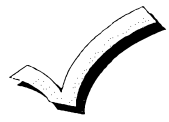
_____

Question 7
Why does the LED stay on when the clock source's frequency is set to a high enough frequency?

_____

_____

_____



The teacher needs to check your work at this point...

Part B - The seven-segment display

27. Wire the circuit of Figure 2.

    Note: Pinout data for the IC and the seven-segment display are at the back of the book.

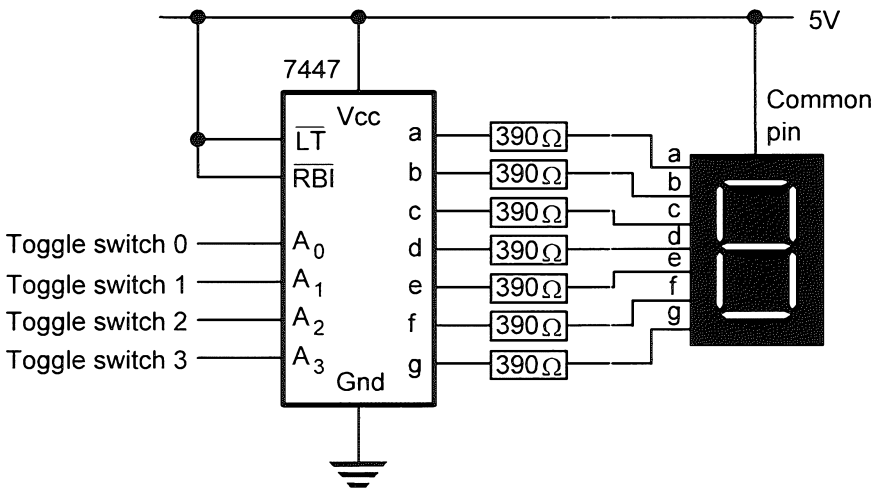


Figure 2

28. Turn on the digital trainer and set the toggle switches so that the input to the 7447 is 0000.

29. Observe what is displayed on the seven-segment display. Record this in the appropriate row in Table 3 (on the next page).

30. Set the toggle switches to the next number in the row and record what is displayed on the seven-segment display.

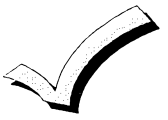
31. Repeat the previous step until the table is full.

Table 3

Switch settings	Display	Switch settings	Display
0000		1000	
0001		1001	
0010		1010	
0011		1011	
0100		1100	
0101		1101	
0110		1110	
0111		1111	

Question 8
What is the relationship between the inputs to the 7447 (connected to the toggle switches) and the number displayed on the seven-segment display?

Question 9
Is the input to the 7447 binary or BCD? Explain your answer using the data in Table 3.



The teacher needs to check your work at this point...

32. Set the IC's input to display the number four.

33. Use a logic probe to measure the logic levels on the 7447 IC's outputs (g to a). Record these in the appropriate cells in Table 4.

Table 4

g	f	e	d	c	b	a

Question 10
Based on the information in Table 4, what logic level on the 7447's outputs turns segments on?

Question 11
Explain why does this logic level turn the segments on? Your answer must include a reference to whether this seven-segment display is common anode or common cathode.

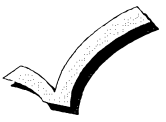
Question 12
What code on the 7447's **output** would turn all of the segments on?

34. Set the 7447 IC's BCD input to 1111.

35. Disconnect the 7447's $\overline{LT}$ input from the +5V supply rail and make it a logic-0 (by connecting it to common). Note what happens on the display.

36. Change the BCD input to several different codes and note what happens on the display.

Question 13
What could this 7447's input be used for and what do you think the letters $\overline{LT}$ stand for?



The teacher needs to check your work at this point...

Copyright: Not to be reproduced without permission

Student notes

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. Draw the schematic symbol for the LED and label the leads.

Questions 2 to 6 refer to the circuit in Figure 1 below

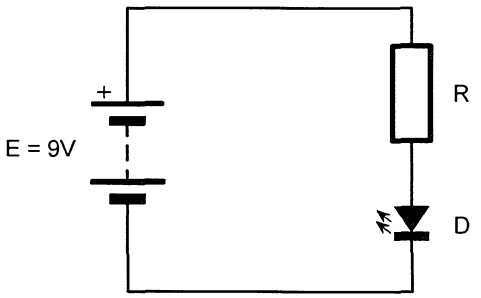


Figure 1

2. What is the purpose of the resistor in the circuit?

 Copyright: Not to be reproduced without permission

3. What value should the resistor be if the LED's operating voltage is 2.2V and the nominal operating current is 12mA?

4. What is the nearest E12 preferred resistor value to your answer to the previous question?

5. What would happen if a 680Ω resistor was used instead?

6. What would happen if the resistor that was used was too big?

Questions 7 and 8 refer to the circuit in Figure 2

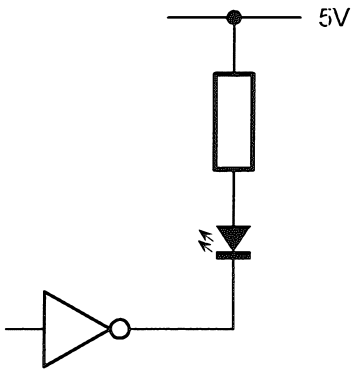


Figure 2

7. What logic level on the NOT gate's **input** would cause the LED to light up?

8. What value should the resistor be if the LED's operating voltage is 1.8V and the operating current is 20mA? (Assume the NOT gate's V_{OL} is 0V.)

9. Name two pieces of equipment that use LED seven-segment displays.

10. How many LEDs are there in a seven-segment display?

11. What segments would be turned on if a seven-segment display was showing the number 4?

12. What is the difference between a common anode and common cathode LED seven-segment display?

13. What must the common pin of a common anode LED seven-segment display be connected to in order to turn on any of the LEDs?

14. What logic level is needed on the segment pins of a common anode LED seven-segment display to turn the segments on?

15. What must the common pin of a common cathode LED seven-segment display be connected to in order to turn on any of the LEDs?

16. What logic level is needed on the segment pins of a common cathode LED seven-segment display to turn the segments on?

17. What is the function of decoder/driver chips?

18. State one advantage and one disadvantage of LED seven-segment displays compared to liquid crystal displays.

19. Name two pieces of equipment that use LCD seven-segment displays.

20. Name the five layers of a reflective-mode LCD.

_____

_____

_____

21. When light bounces off the mirror, it is not allowed to leave a reflective-mode LCD

- ☐ in the regions where the segment electrodes are active.
- ☐ in the regions where the segment electrodes are inactive.
- ☐ across the entire length of the display regardless of whether segment electrodes are active or inactive.

22. Which of the five layers is not used in a transmissive-mode LCD?

_____

23. A segment electrode is made active by

- ☐ making the segment electrode logic-1 and the backplane electrode logic-0.
- ☐ making both the segment electrode and the backplane electrode logic-1.
- ☐ making the segment electrode logic-0 and the backplane electrode logic-1.
- ☐ connecting anti-phase squarewaves to the segment electrode and the backplane electrode.

 Copyright: Not to be reproduced without permission

Questions 24 and 25 refer to the circuit in Figure 3 below

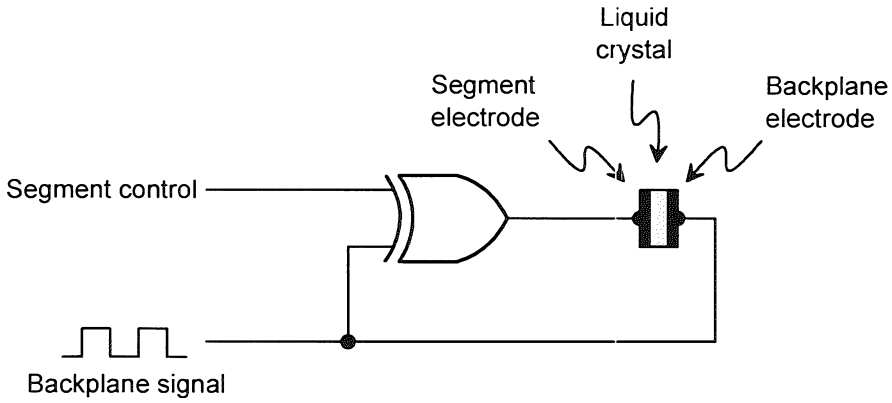


Figure 3

24. What logic level on the segment control input would activate the segment and turn it on (that is, make it visible)?

25. Why is there anti-phase squarewaves on the segment and backplane electrodes when the segment control input is a logic-1?

26. State one advantage and one disadvantage of LCD displays compared to LED seven-segment displays.

Section 9 Interfacing techniques

Purpose To develop an understanding of some of the basic techniques used to interconnect digital (and microprocessor based) control systems to the electrical devices used to control processes.

Objectives At the end of this section you should be able to:

- Describe the need for *interfacing* circuitry between digital control systems and electrical input and output devices
- Identify simple BJT/MOSFET buffer/driver circuits used to control a DC load from a logic device and explain their operation
- Choose suitable replacement components for a faulty BJT/MOSFTET buffer/driver circuit
- State two devices that can be used to provide electrical isolation between the output of logic devices and high voltage/current loads
- Identify simple circuits that provide electrical isolation between the output of logic devices and high voltage/current loads and describe their operation
- Identify simple circuits used to connect switches to the input of logic devices and describe their operation
- Describe *switch bounce* and explain the problems it can cause when switches are used as input devices to digital devices
- Identify simple RC circuits used to implement debouncing and explain their operation together with a logic input with hysteresis
- Identify simple comparator circuits used to interface transducers to the input of logic devices and explain their operation
- Use a DMM to test and measure interfacing circuits between logic and I/O devices

Introduction

An important application of digital electronics is the control of equipment. Using microprocessors and microcontrollers, digital electronics controls the operation of white goods like washing machines, dishwashers, etc as well as equipment like air-conditioning systems, alarm systems, other more sophisticated access control systems, car engines, factory production lines, and much, much more.

A requirement of most digital electronic control systems is the ability read input devices (eg switches, keypads, sensors, etc) and use that formation to turn on and off output devices (LEDs, lamps, buzzers, siren, relays, solenoids, motors, etc). Importantly, most input and output devices cannot be directly connected to digital devices for a variety of reasons. This chapter explores why this is the case and looks at basic interfacing circuitry that can be used.

Connecting relatively high current devices to the output of digital devices

Although basic theory would suggest that the two circuits below in Figure 1 would work, (after all, you can control LEDs using current sinking this way) the fact is, neither of them can be used.

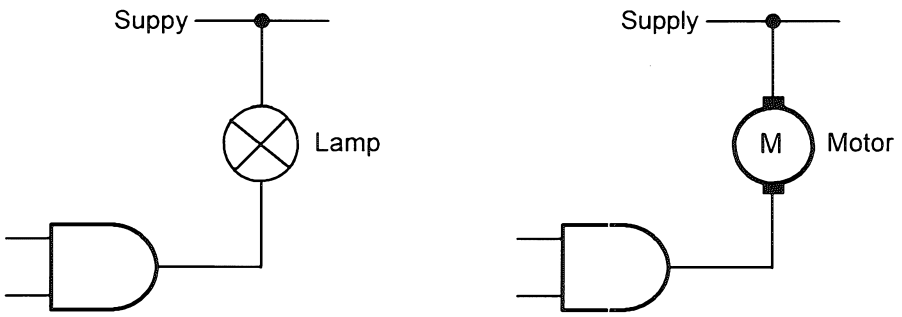


Figure 1 *Circuits that cannot be used to control DC loads like lamps and motors*

There are two reasons that prevent these circuits from working. First, although there are DC lamps that can draw as little as 10mA, the vast majority of them have operating currents over 30mA. And, the operating current of DC motors starts at 150mA but they're typically much higher than this. Importantly, most logic devices don't have the ability to sink these types of currents (recall that the 74AS series TTL has the highest I_{OL} at 20mA). Second, many output devices don't operate on TTL nor CMOS voltage levels. That being the case, a circuit must be used to translate between the electrical requirements of the logic device and the output device.

The BJT buffer/driver

The BJT buffer/driver shown in Figure 2 is a simple interfacing circuit that is widely used. It is shown here with a lamp but it can be used for other DC loads like motors and relays.

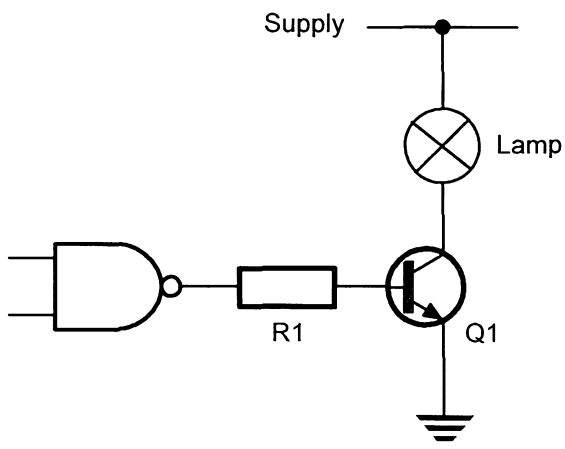


Figure 2 *A BJT buffer/driver acting as an interface between the logic device and a lamp*

In this circuit, the BJT is being used as a switch; that is, it is only operated in cut-off (no collector current) or at/near its maximum operating current (ie the transistor is driven into saturation).

Operation

- When the logic gate's output is logic-0 (which is 0V) there is no potential difference across the BJT's base resistor (*R1*).
- This means that there is no base current and consequently no collector current (that is, the transistor is cut-off). Therefore the lamp is off.
- When the logic gate's output is logic-1 (5V) a potential difference across the base resistor is developed.
- This means that there is a base current and consequently a collector current. Therefore the lamp is on (assuming the circuit has been designed correctly).

Before we analyse a circuit with component values, notice that the lamp doesn't have a current-limiting resistor connected in series with it. This is because DC loads such as lamps, motors and relays usually have a DC resistance. This means that they are self-limiting in terms of their operating current requirements and so a resistor is unnecessary. Another important feature of the circuit is that the load can be connected to any supply voltage - it doesn't have to be connected to the logic device's supply rail.

Figure 3 below shows the same circuit as Figure 2 but with component values for analysis.

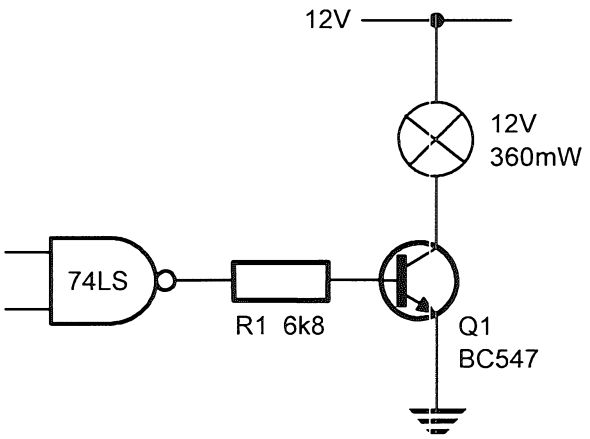


Figure 3 *The circuit in Figure 2 with component values*

The lamp's electrical specifications can be used to determine its operating current and therefore the BJT's collector current.

$$I_C = I_L = \frac{P_L}{V_L}$$

$$I_C = I_L = \frac{360mW}{12V}$$

$$I_C = I_L = 30mA$$

The minimum base current required to permit a 30mA collector current can then be calculated.

$$I_B = \frac{I_C}{\beta}$$

$$I_B = \frac{30mA}{100}$$

$$I_B = 300\mu A$$

Importantly, the base current can be larger than this (ensuring the BJT is driven into saturation) but not lower.

Next, the value of $R1$ can be checked to ensure that it's appropriate (if it's too big, it wont let 300µA of base current flow which in turn would limit the collector current).

$$I_{R1} = \frac{V_{R1}}{R1}$$

$$I_{R1} = \frac{V_{OH} - V_{BE}}{R1}$$

$$I_{R1} = \frac{2.7V - 0.6V}{6k8}$$

$$I_{R1} = 309\mu A$$

Notice that the logic gate's minimum output for a logic-1 (V_{OH}) is used for this calculation and not the standard voltage for a logic-1. This is to ensure that the circuit is designed for a worst case scenario. It's for the same reason that a beta of 100 was chosen (the beta of the BC547 ranges from 100 to 800). If the actual figures for beta and logic gate output voltage are higher, this won't hamper the circuit's performance.

Next, when the logic gate's output is logic-1 (turning the lamp on), the gate will be current sourcing the BJT's base current which we've established is required to be at least 300µA. Knowing this, the adequacy of the logic gate's source current (I_{OH}) can also be checked. A quick look at the Table 4 on page 7-11 shows that the 74LS can source up to 400µA so this design should work.

Another circuit parameter that can be checked is the BJT's power dissipation. This can be found by adapting the equation $P = I \times V$ to obtain $P_{Q1} = I_C \times V_{CE}$. The collector current was calculated earlier and, if the circuit is designed well, the BJT will be driven into saturation giving a collector-emitter voltage of up to 0.6V max. That being the case, the transistor's power dissipation is:

$$P_{Q1} = I_C \times V_{CE}$$

$$P_{Q1} = 30mA \times 0.6V$$

$$P_{Q1} = 18mW$$

As the maximum power dissipation of a BC547 is 500mW, this BJT is appropriate for the application.

Replacing faulty components with substitutes

Although you're not expected to design circuits such as Figure 3, as a technician may be required to repair one. Moreover, for the reasons discussed in Section 7, you may also be forced to replace faulty component in such circuits with substitutes. Importantly, issues that inform the circuit's design will also inform your device substitution decisions.

When the transistor is faulty:

- The substitute's rated values of $I_{C(max)}$ and $V_{CE(max)}$ should be the same as the original part or greater that the circuit's load current and supply voltage. If not, the substitute may be damaged.
- The power rating of the substitute must be the same as the original part or greater than the maximum power it will dissipate in the circuit. If not, the substitute may be damaged.
- The substitute's minimum beta (β) must be the same or greater than the original part. If not, the logic gate's maximum source current may not be sufficient to drive the transistor.

When the resistor is faulty:

- The substitute must be the same value or slightly lower than the original part. If not, the BJT's base current will be insufficient to produce the necessary collector/load current.
- The substitute must have the same power rating as the original part or greater. If not, the substitute may be damaged.

When the logic gate is faulty:

- The substitute must be from the same sub-family or a faster sub-family. If not, the logic gate's maximum source current may not be sufficient to drive the transistor. (The faster sub-families tend to be able to source and sink higher currents.)

Other buffer/driver circuits

Problems arise when the logic gate is not able to source sufficient current to the base of the BJT to allow the appropriate amount of collector/load current. This usually occurs when either: a) the load current is so large that the BJT's required base current is beyond the logic gate's source current capability; or b) the appropriate BJT's beta is too small to be able to amplify the source current (these two issues often combine). There are several solutions to this problem that are used.

Darlington pair transistor buffer/driver

One buffer design uses a *Darlington pair* transistor instead of a single transistor as shown in Figure 4 below. Darlington pairs can be made by simply connecting two transistors as shown but you're more likely to see them as a single three pin package.

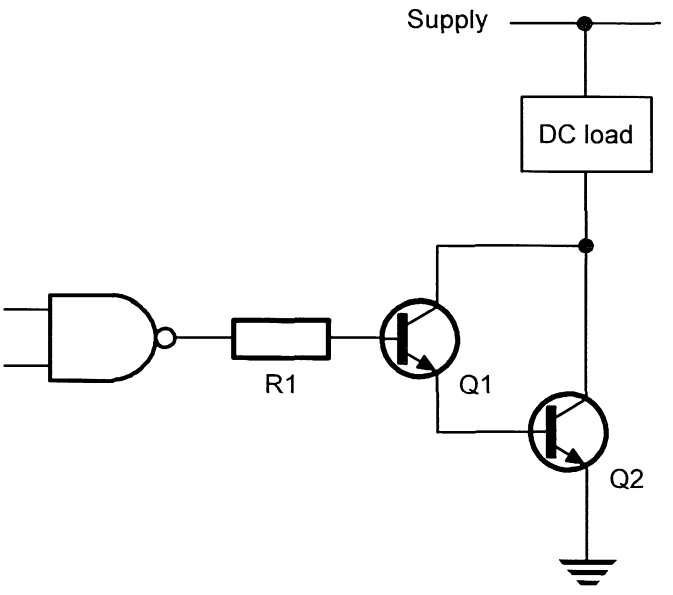


Figure 4 *Darlington pair BJT buffer/driver*

The advantage of the Darlington pair is that the total beta is equal to the multiplication of the beta of the individual transistors ($\beta_T = \beta_{Q1} \times \beta_{Q2}$). This can provide for very large values of beta which would readily allow the source current of low power TTL sub-families and even CMOS logic gates to provide sufficient base current. You'll learn more about Darlington pairs in another subject.

MOSFET buffer/driver

Another widely used buffer design uses an enhancement-mode MOSFET in place of the BJT as shown in Figure 5 below.

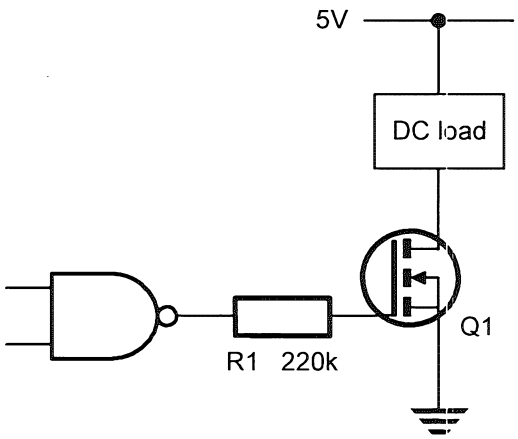


Figure 5 *An enhancement-mode MOSFET buffer/driver*

The advantage of using a MOSFET is that the circuit's operation becomes largely independent of the logic family and sub-family of the gate driving it. To understand why, recall that FETs are voltage-operated devices; that is, a voltage on the gate controls the current through the drain and source (recall that the gate current is virtually zero). Consequently, the current sourcing capabilities of the logic gate in the circuit are irrelevant.

Operation

The operation of this circuit is basically the same as the operation of the BJT buffer but for a slightly different reason.

- When the output of the logic gate is a logic-0 (which is 0V) the voltage on the gate is 0V.
- This means that the MOSFET's V_{GS} is 0V and the drain-source current is zero. Therefore the lamp is off.
- When the output of the logic gate is a logic-1 (5V) the voltage on the gate is 5V.
- This means that the MOSFET's V_{GS} is 5V and so drain-source current would flow. Therefore the lamp is on.

Notice that the gate resistor (*R1*) can be a much higher value than the base resistor in the BJT equivalent circuit as there is no gate current to be set.

Electrical isolation

When the input or output device connected to a digital circuit uses voltages and/or currents that can potentially destroy the electronics or even harm people, its necessary to electrically isolate the I/O device from the circuit. There are several devices that can provide the required electrical isolation:

- relays
- optocouplers
- solid state relays

Relays

A relay is basically an electromagnetically-operated switch. The electromagnetic is just a coil and the switch is spring-loaded to set the switching arm to a default position. Switch options in relays are the same as for manual switches (ie SPST-N/O, SPST-N/C, SPDT, DPST, DPDT and so on). Some schematic symbols used for relays are shown in Figure 6 below. The physical size of relays varies and is affected by the power rating of the device. Sizes can range from tiny PCB mounted types to larger screw-terminal varieties.

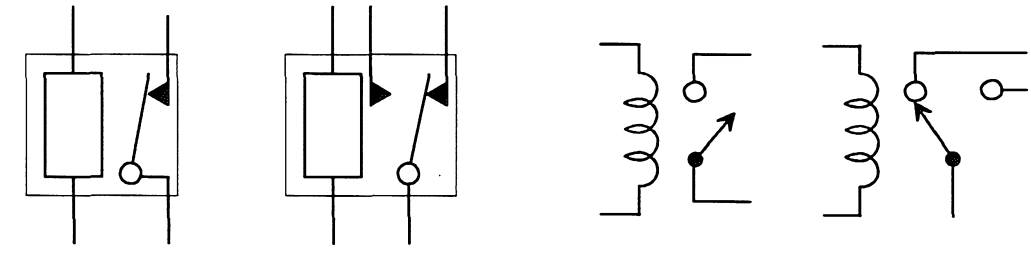


Figure 6 *Schematic symbols used to represent relays*

When there is no current flowing through the relay's coil it's said to be *de-energised* and the switch is in its rest or default position. When a current is passed through the coil, an electromagnetic field builds up around it and the coil is said to be *energised.* The electromagnetic field attracts the magnetic switch, causing it to be thrown. When the coil is subsequently de-energised the spring pulls the switch to its default position.

Importantly, there is no electrical connection whatsoever between the coil circuit and the switch circuit. This means that the two circuits are electrically isolated from each other and this isolation can be rated up to several thousand volts depending on the type of relay.

The energise current of most relays is beyond the sink/source current capabilities of logic gates so one of the buffer/driver circuits must be used as an interface. Figure 7 below the BJT buffer/driver introduced earlier controlling a relay which in turn is used to control the operation of a 240V AC motor.

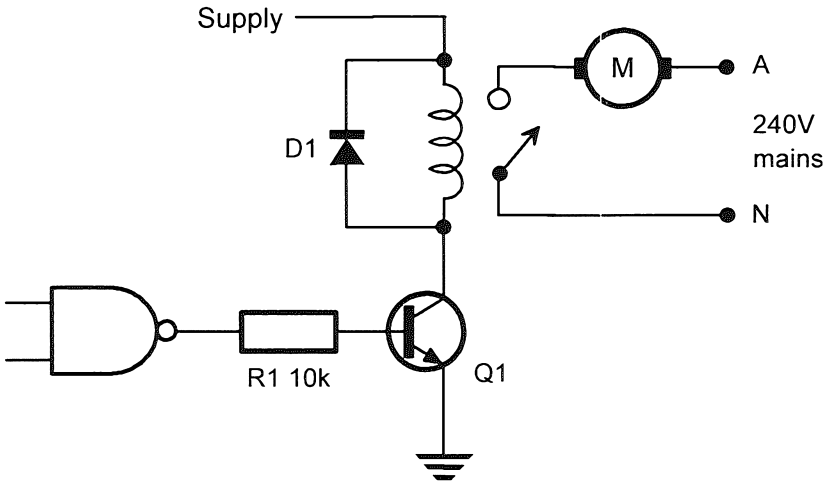


Figure 7 *BJT driver circuit used to interface a logic gate to a relay which in turn controls an AC motor*

Operation

When the output of the gate is logic-0:

- The transistor is turned off
- So the relay is de-energised
- Which means the switch is open
- And the motor is off

When the output of the gate is logic-1:

- The transistor is turned on
- So the relay is energised
- Which means the switch is closed
- And the motor is on

A collector resistor is not necessary as the DC resistance of the coil will "self-limit" the collector current because base bias is used. This resistance will be different from relay to relay but is typically in the order of 300Ω to several kilo ohms.

Note that, because the relay's coil is an inductive load, a *commutating diode* (*D1*) must be connected across it in the polarity shown. This effectively short-circuits the large voltages that would otherwise develop across it as it is de-energises. Such voltages could easily exceed the BJT's V_{CE} and damage it.

Although relays can provide excellent electrical isolation characteristics, they have several disadvantages which make them unsuitable in some applications:

- Their moving parts makes them prone to mechanical wear and failure
- Relays are only suitable for slow speed switching (with the higher powered relays being the slowest of all because it takes time to get that mass of metal in the switch moving)
- The switch contacts inside the relay bounce causing the contacts to arc which produces contact wear (by burning them) and is dangerous in mining and petro-chemical industries (the arcing could trigger an explosion)

Optocouplers

Relays are electro-mechanical devices but optocouplers are purely electronic (semiconductor) devices. They are most commonly available in 4 pin, 6 pin and 8 pin IC packages both in DIP and SMD packages.

Optocouplers consist of an LED and a photosensitive component of some sort. Figure 8 shows a typical example of an optocoupler using a photo-sensitive transistor but other photo-sensitive components such as diodes, FETs, Darlington pairs, SCRs, Triacs and even logic gates are used in place of the transistor.

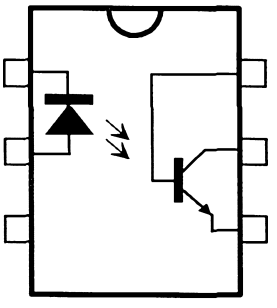


Figure 8 *A typical IC optocoupler*

Electrical isolation is achieved by the fact that the two devices inside the chip are physically separated. Light is used to allow the diode to activate the photo-sensitive device. For example, the phototransistor in the IC above has a base that can be forward biased like any normal transistor (that is, by making its V_{BE} 0.6V), but it can also be forward biased by the light from the LED (the IC is fully sealed so incidental light wont affect its operation).

The typical characteristics of optocouplers include:

- LED power dissipation of 200mW
- Electrical isolation between the components of 1,500V
- Maximum output device power dissipation of 200mW
- Operating temperature from -55°C to +100°C

An example of an interface circuit using the kind of optocoupler in Figure 8 is shown below in Figure 9.

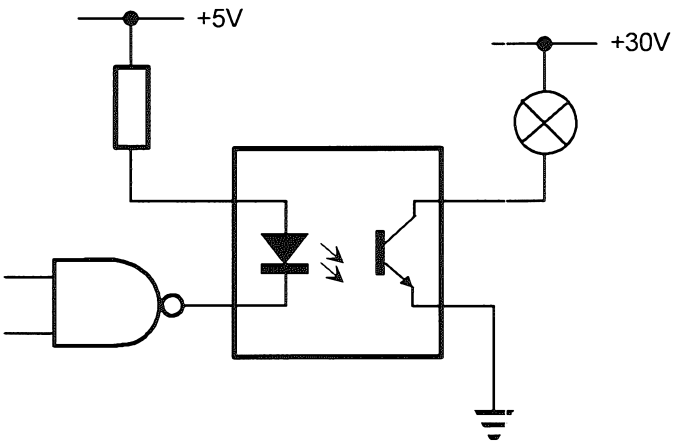


Figure 9 *A typical interface circuit using an optocoupler*

This arrangement involves two circuits that you have already seen. The phototransistor is wired to operate as the BJT buffer/driver introduced earlier in this section. The LED is operated by a logic gate and the circuit is wired so that it *sinks* the LED current as introduced in an earlier section.

Operation

When the output of the gate is logic-1:

- There is no potential difference across the LED and *R1*
- So, no current flows and the LED is off
- Without light, the transistor is reverse biased so no collector current flows
- So the lamp is off

When the output of the gate is logic-0:

- There is a potential difference across the LED and *R1*
- So, current flows and the LED is on
- With light, the transistor is forward biased and a collector current flows
- So the lamp is on

Optocouplers have several advantages over relays:

- There are no moving parts so there is no mechanical wear
- They can switch at much higher speeds
- There there are no contacts so there are no arcing problems

However, although optocouplers provide good electrical isolation characteristics (up to 1,500V), relays can have much higher isolation ratings and can switch much higher currents.

Solid state relays

Solid state relays are another form of optocoupler but used for much higher voltage and current applications. They have the same advantages as other optocouplers but they are much more expensive than relays. As such, they tend to be used in *flame-proof* applications such as mines, petro-chemical facilities and so on.

Connecting devices to the input of digital circuits

Connecting switches to inputs

The simplest way to interface a two-position switch to the input of a digital circuit is to wire it so that in one position the switch is connected to ground and in the other position it is connected to 5V (or V_{DD} if CMOS is being used). This is shown in Figure 10 below.

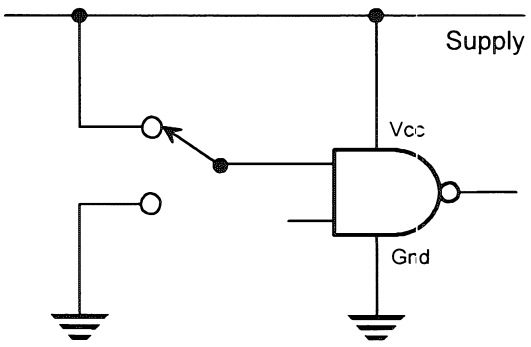


Figure 10 *A two-position switch connected to the input of a logic gate*

However, there is a problem with this basic arrangement and so it's unlikely to be used in digital control systems.

To explain, consider the moment when the switch is thrown. The switch's schematic symbol shows that there must be a moment when the switch is not connected to either of its two contacts. During this time, the switch's "pole" (ie the contact connected to the moving arm) is "floating" and so too must be gate's input is. While the gate's input voltage will likely be near 0V, it'll drift around by hundreds and even thousands of millivolts in response to electromagnetic fields emitted by nearby conductors and from multiple other human-made and natural sources.

Floating voltages on the inputs to logic gates are highly undesirable as they can potentially be sufficiently large to fall in the gate's transition zone and then randomly interpreted as either logic level. This causes spurious logic levels on the output of the logic gate during this time which may adversely affect the performance of the whole circuit (depending on what the circuit is).

To solve this problem, a slightly different switch arrangement to the one shown in Figure 10 is used. It involves a single position switch instead of a two position switch and examples are shown in Figure 11 on the next page.

Figure 11a shows how the switch provides a logic-0 to a TTL logic gate input that becomes a logic-1 the moment the switch is thrown. Figure 11b shows the same switch but with reversed logic levels. In both cases, the logic gate's input never floats.

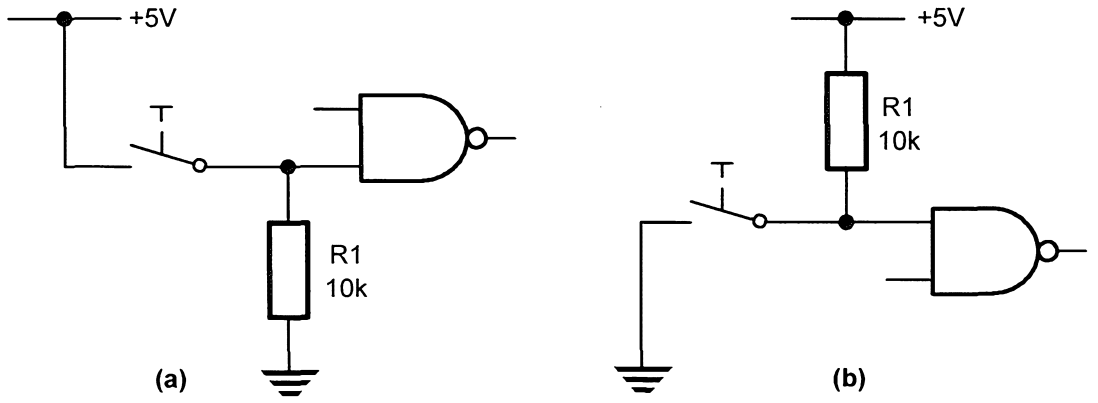


Figure 11 *When thrown, the switch will: (a) change the input from a logic-0 to a logic-1 and; (b) change the input from a logic-1 to a logic-0.*

This is a great solution but there's another problem with switches connected to logic gate inputs that may need to be addressed. It's called *switch bounce.*

Switch bounce

Switch bounce occurs whenever any mechanical switch is thrown. To describe this using the single position switches shown above, a continuous electrical connection isn't obtained when the switch's contacts first meet. This is because the moving contact bounces off the fixed contact at least a few times before settling. This means that, instead of producing a clean 0-to-1 or 1-to-0 transition on a logic gate's input (as shown in Figure 12a), the bouncing contacts producing several transitions before settling as shown in Figure 12b. The "signal" produced is often called *chatter.*

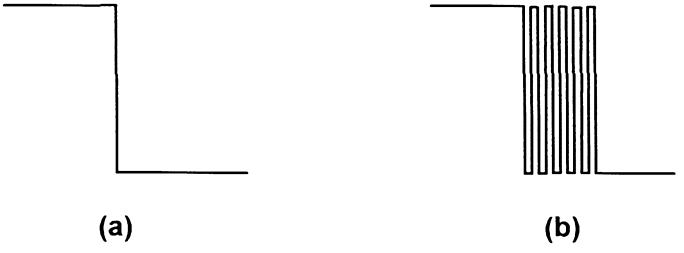


Figure 12 *(a) An example of the ideal response when a switch is thrown and; (b) the actual response due to switch bounce*

Switch bounce typically lasts up to 10ms and, for a practical example of the problem it can cause, consider a computer keyboard. When the key for the letter "k" is pressed and released the chatter from the switch bounce would simulate the key having been pressed many times in quick succession and produce a string of k's on the computer screen instead of a single instance of it.

Methods for preventing these multiple transitions from being generated are called *debouncing*. Switch debouncing for computer keyboards and keypads are usually managed by the software these days. However, the switches in simpler systems still rely on hardware debouncing. There are several methods used with an easy to understand example involving an RC circuit and using a logic gate with an in-built *hysteresis* effect on its inputs as shown in Figure 13 below.

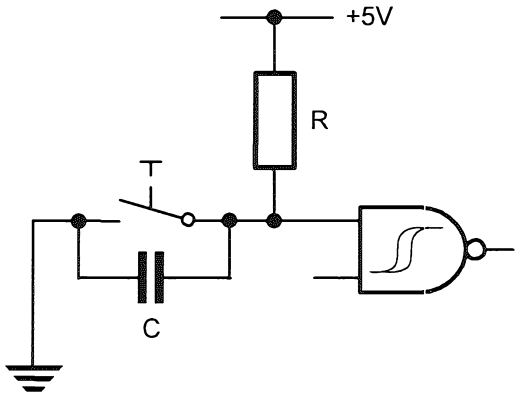


Figure 13 *Switch debouncing is provided for by using an RC circuit together with gate that has inputs with an in-built hysteresis effect*

The term *hysteresis* is borrowed from AC theory and describes the manner in which an iron core of a transformer is magnetised-demagnetised during alternating halves of a sinewave. We wont go into an explanation of that here except to say that the term has been borrowed to describe an action where the trigger point for a reaction in one direction is different to the trigger point that causes a reaction in the reverse direction.

To give you a practical example of hysteresis, have you ever wanted to cook a frozen pizza and have been in a hurry? The oven seems to take forever to heat up doesn't it? To check the oven's progress you've probably turned the temperature control down until you've heard a click and the oven light goes out. This is the thermostat turning off and it gives you an idea of how close the oven is to the temperature you want. But have you ever noticed that when you turn the temperature control back up to the required setting the thermostat doesn't click on at the same point it clicked off? It turns back on at a higher temperature setting. This is the hysteresis effect at work. (As an aside, hysteresis is intentionally built into oven temperature control systems to prevent an effect called *hunting*. Hunting occurs when the system turns on and off at short intervals as the temperature increases just above the preset level then drops just below it.)

To appreciate the difference between a normal logic gate and one with hysteresis, let's compare the two. Consider a normal NAND gate first. Recall that the V_{IL} of 74ALS series TTL logic devices is ≤0.8V and the V_{IH} is ≥2V. That being the case, Figure 14 below shows how a normal 74ALS series NAND gate would interpret a 0V-5V-0V ramp on its inputs.

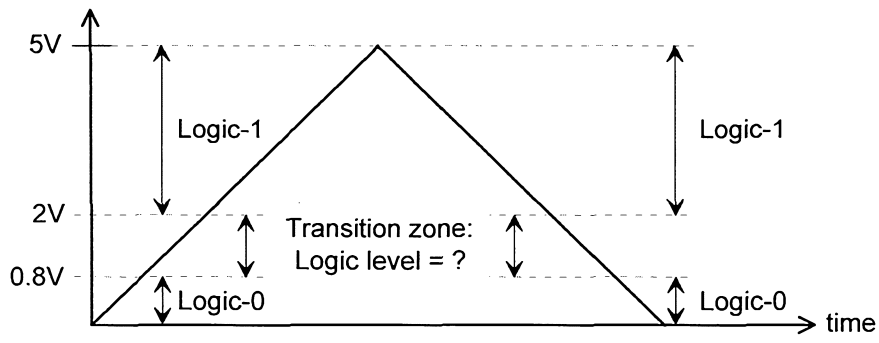


Figure 14 *The response by the inputs of a normal 74LS TTL series NAND gate to a 0V-5V-0V ramp*

There are probably no surprises for you in Figure 14! However, thinking about the issue this way serves as a useful reference point when considering how the inputs to a logic gate with hysteresis are read. This is shown in Figure 15 below.

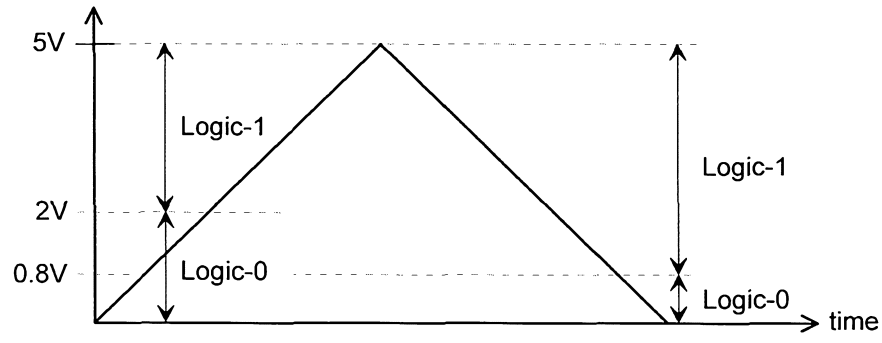


Figure 15 *The response by the inputs of a 74LS TTL series gate with hysteresis to a 0V-5V-0V ramp*

As you can see, logic gate inputs with hysteresis don't have a transition zone. The voltages between 0.8V and 2V (the transition zone of normal gates) are considered a logic-0 if the input was previously a logic-0 and as a logic-1 if the input was previously a logic-1.

Now, to understand how the series RC network, together with the gate's hysteresis effect, can minimise the effects of switch bounce, consider the circuit of Figure 15 again redrawn below in Figure 16.

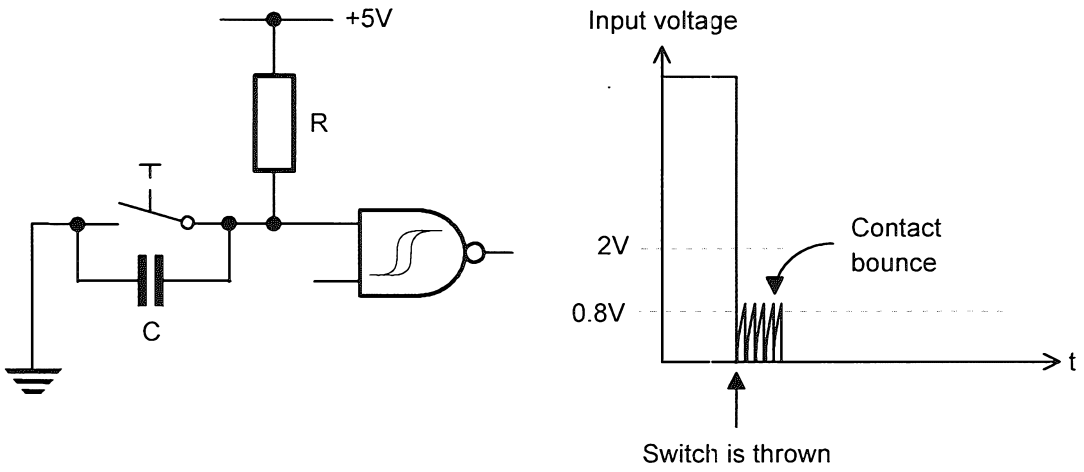


Figure 16

Suppose the switch is the open position and the capacitor is fully charged, rendering a logic-1 on the gate's input. When the switch is closed, the capacitor would be instantly discharged causing the logic gate's input to register the zero volt potential difference across it as a logic-0. In turn, this would set the threshold voltage required on the gate's input to be 2V or more to be accepted as a logic-1.

Then, when the switch contact bounces, the open-circuit across the capacitor would cause it to start charging towards 5V. But, importantly, it does so rising exponentially rather than instantaneously. If the time constant of the series RC network is sufficiently large (though still only in the order of milliseconds), the capacitor's potential difference will not reach 2V before the switch contacts make again after the bounce, restarting the capacitor's charge from 0V again for the subsequent bounce. The graph on the right of Figure 16 shows how the size of the bounce voltages are suppressed and remain below the threshold on the input for a logic-1 even if it exceeds 0.8V. Neat?

Connecting transducers to inputs

A variety of transducers are used in digital control systems to "sense" the system's condition. Examples include; photodiodes, phototransistors & LDRs (light), thermistors & thermocouples (heat), hall-effect devices (magnetism), photoelectric devices (smoke), piezoelectric devices (tension or "strain"), and many more.

Most transducers are not sources of EMF - that is, they don't generate voltages. Those that do (like the thermocouple and piezoelectric strain gauge) only produce small voltages that well below TTL and CMOS voltages for a logic-l. For these reasons, transducers must be connected to digital systems via an appropriate interfacing circuit.

There are many circuit designs that can be used for this purpose but we'll consider a classic one which you already know a little about; the comparator. In this circuit example, an LDR is used to detect light and the comparator is used to determine whether the light has crossed a threshold level on its way to getting lighter or darker. The circuit is shown in Figure 17 and can be adapted for use with other transducers.

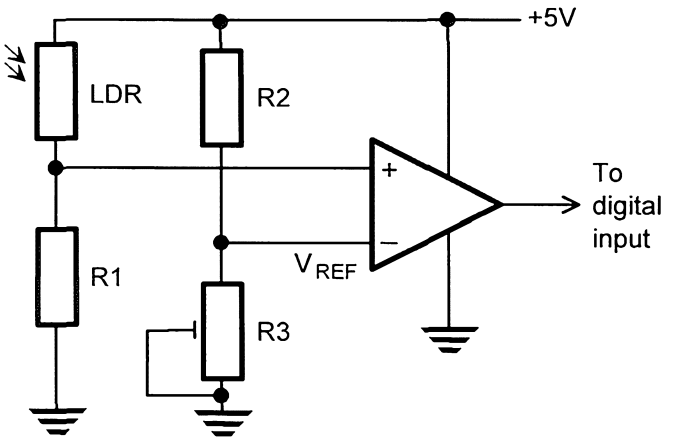


Figure 17 *An interfacing circuit for a transducer using an inverting comparator*

The op amp doesn't have any negative feedback giving it an extraordinarily large gain. In this condition, the op amp is useless as a signal amplifier but makes a perfect comparator. Recall that comparators output only two voltages, *Vcc* and *-Vcc* (so, 5V and 0V in the case of the single rail op amp in Figure 17).

The rule that governs the op amp's output voltage is simple. When the op amp's non-inverting pin is a more positive voltage than its inverting pin, the output is 5V. When the op amp's non-inverting pin is a less positive voltage than its inverting pin, the output is 0V.

The resistors *R2* and *R3* form a voltage divider with the potential difference across *R3* setting the op amp's inverting input to a fixed voltage that serves as the comparator's reference voltage (V_{REF}).

The LDR and resistor *R1* also form a voltage divider with the potential difference across *R1* setting the voltage on comparator's non-inverting input which is compared to the reference the reference voltage. Recall that, in the absence of light, an LDR's resistance very large but the presence of light, it decreases .

Now, when the ambient light is sufficiently bright, the resistance of the LDR relative to *R1* in this circuit would be so small that the potential difference across *R1* would exceed the reference voltage. Under these conditions, the comparator's output would approach 5V and be accepted as a logic-1 by the digital circuit that its connected to (which would act on that according to the purpose of the circuit, for example, turning some street lights off).

But as the ambient light goes down, the resistance of the LDR increases, obtaining a greater share of the supply voltage and, in turn, reducing the potential difference across *R1*. Once the ambient light is low enough, the potential difference *R1* drops below the reference voltage. Under these conditions, the comparator's output would approach 0V and be accepted as a logic-0 by the digital circuit, and acted on accordingly (for example, turning some street lights on).

The resistor *R3* in Figure 17 is a trimpot to allow the threshold point at which the ambient light causes a change in logic level to be adjustable.

Hunting

There is a problem with the circuit in Figure 17 if its used in applications like turning on and off street lights. As the sun sets, and so the ambient light nears the threshold, a passing cloud could temporarily lower the light levels enough to cause the voltage on the comparator's inverting input to drop below V_{REF} and causing the comparator to turn the lights on. But, once the cloud passes, the ambient light level could increase enough to cause the voltage on the comparator's inverting input to go back above V_{REF} and causing the comparator to turn the lights back off. This is another example of hunting - It's fairly inelegant and definitely undesirable to have the street lights turning on and off due to temporary environmental factors at dawn and dusk.

To prevent hunting in situations like this, a comparator with hysteresis (called the Schmitt Trigger) can be used. The Schmitt Trigger is shown in Figure 18 on the next page but its operation isn't discussed here. Hunting is prevented because, when the comparator's output changes, the reference voltage changes too and so the voltage on its inverting input must change by a substantial amount to cause it to change back.

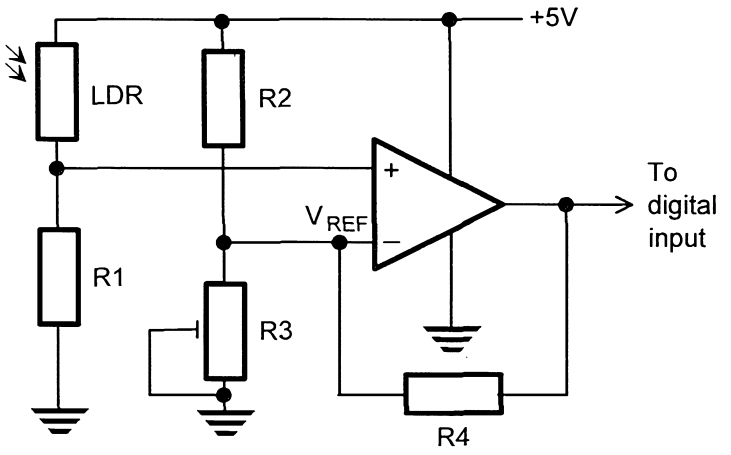


Figure 18 *An interfacing circuit for a transducer using a comparator with hysteresis (the Schmitt Trigger)*

Copyright: Not to be reproduced without permission

Student notes

Skill practice 9

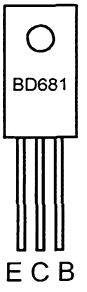
An introduction to testing and measuring digital interfacing circuits

This exercise is practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 1¼ hours to complete this exercise.

Equipment

- Digital trainer
- DMM
- 12V lamp panel
- 74LS04
- BD681 transistor
- 4N25 (or 4N28)
- 12V DC PCB relay (JW25N)
- 3k9Ω ¼W resistor
- 2k7Ω ¼W resistor
- 1N4004 diode
- hook-up wire
- wire strippers

Transistor pinout



Plastic side facing you

Remember:

Follow TAFE NSW WHS guidance at all times!

 Copyright: Not to be reproduced without permission

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

2. Gather the equipment needed for this exercise.

3. Wire the circuit of Figure 1 below.

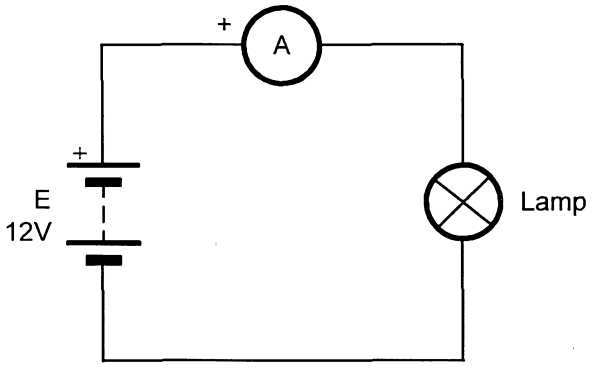


Figure 1

4. Measure and record the lamp's operating current: _____

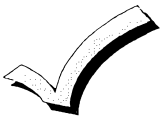
Question 1
Why is it not possible to control this lamp directly from any logic gate using current sourcing or sinking? **Tip:** Refer to the information in the table on page 5-11.

_____

_____

_____

_____



The teacher needs to check your work at this point...

5. Wire the circuit of Figure 2 below.

Note: Data for the 74LS04 is provided on the inside back cover.

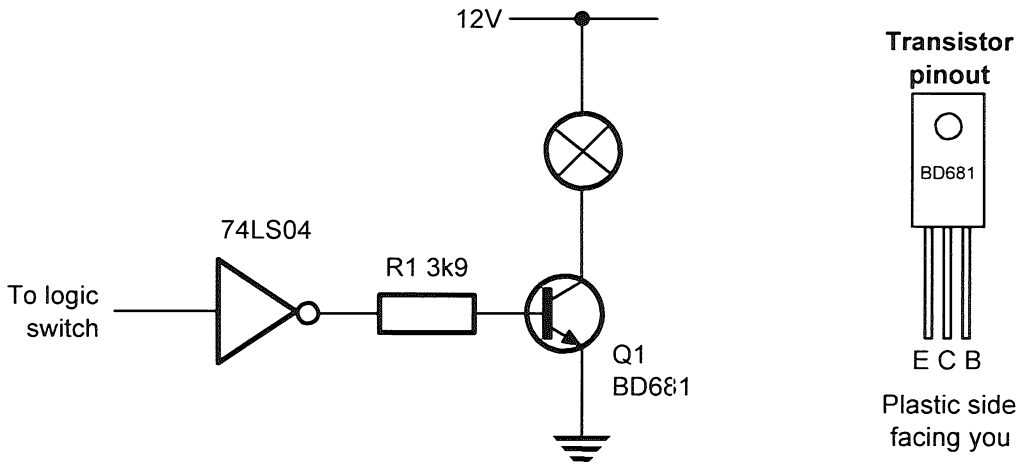


Figure 2

6. Operate the switch on the NOT gate's input and verify that the lamp turns on and off.

Question 2
What logic level on the **input** of the NOT gate turns the lamp on?

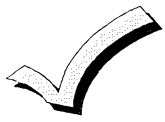
Question 3
Explain how the logic level you gave in your answer to Question 2 turns the lamp on. **Tip:** Your answer should include references to the basic operation of NOT gates and BJTs.

7. Measure and record the transistor's base voltage: _____

Question 4
What does the transistor's base voltage tell you about the BD681?

_____

_____



The teacher needs to check your work at this point...

8. Modify the transistor's collector circuit as shown in Figure 3 below.

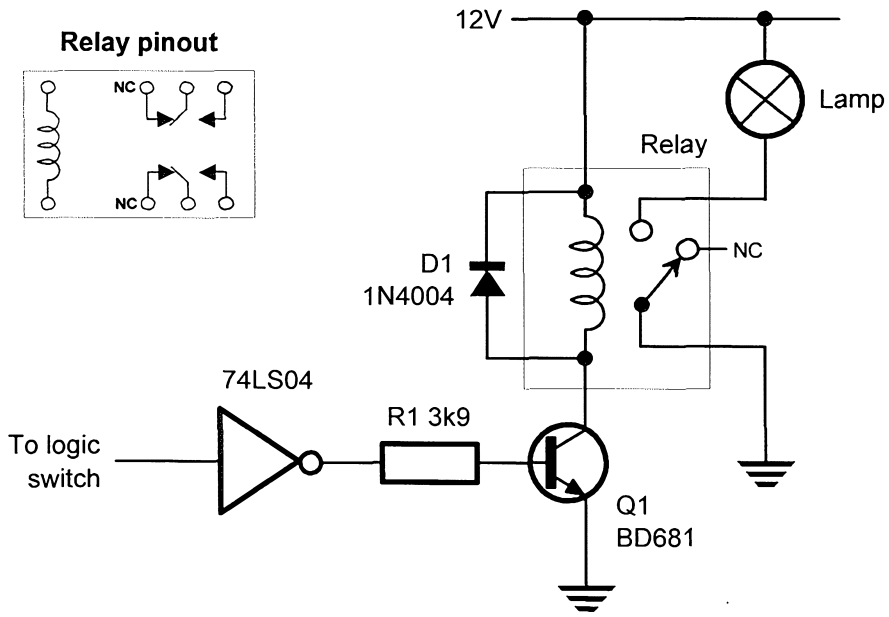


Figure 3

9. Operate the switch and verify that the lamp turns on and off.

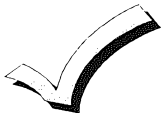
 Copyright: Not to be reproduced without permission

Question 5
What logic level on the **input** of the NOT gate turns the lamp on?

Question 6
Explain how the logic level you gave in your answer to Question 5 turns the lamp on. **Tip:** Your answer should include references to the basic operation of NOT gates, BJTs and the relay.

Question 7
Using the transistor, together with a relay permits control over much greater load currents (amps versus hundreds of milliamps). What other potentially useful feature does the circuit in Figure 3 have that the circuit in Figure 2 doesn't have?

Question 8
What simple modification to the circuit (that wouldn't involve removing or adding any components) would require a logic-1 on the NOT gate's input to turn the lamp on instead of a logic-0?



The teacher needs to check your work at this point...

10. Wire the circuit of Figure 4 below.

    Note: Data for the 4N25 is provided on the inside back cover.

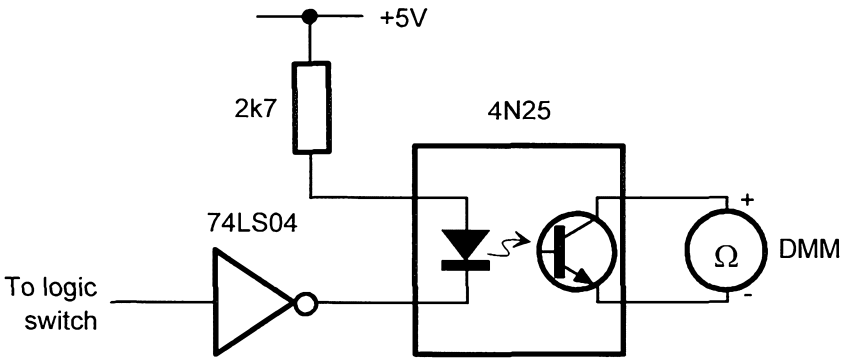


Figure 4

11. Set the NOT gate's input to logic-0.

12. Turn on the trainer and measure the resistance between the BJT's collector and emitter terminals (measure between pins 4 and 5 of the IC). Record this value in Table 1 below.

13. Change the NOT gate's input to logic-1. Measure and record the BJT's new C-E resistance.

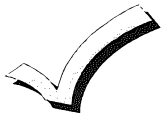
Table 1

C-E resistance when switch is set to a logic-0	C-E resistance when switch is set to a logic-1

Question 9
Briefly explain what's going on around the circuit when you make the input to the NOT gate a logic-0. **Tip:** Your answer should include references to the basic operation of NOT gates, the LEDs and photosensitive BJTs.

Copyright: Not to be reproduced without permission

Question 10
Briefly explain what's going on around the circuit when you make the input to the NOT gate a logic-1.



The teacher needs to check your work at this point...

Review questions

Use these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

1. Give three reasons for using interfacing circuitry between digital electronics and electrical devices used in industrial processes.

2. Draw the circuit of an enhancement mode MOSFET buffer/driver with a 24V 10W lamp for the load and driven by TTL logic gate. Show power supply voltages and component values where appropriate.

 Copyright: Not to be reproduced without permission

Questions 3 to 6 refer to the circuit of Figure 1 below

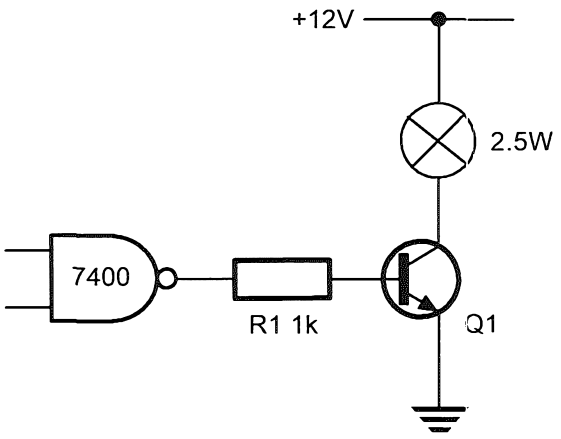


Figure 1

3. What logic level on the output of the logic gate causes the lamp to turn on?

4. Why isn't a collector resistor needed for the circuit?

5. What two *preferred* resistor values could be used if *R1* went open-circuit and you didn't have a spare 1kΩ resistor?

Questions 6 to 9 refer to the circuit of Figure 2

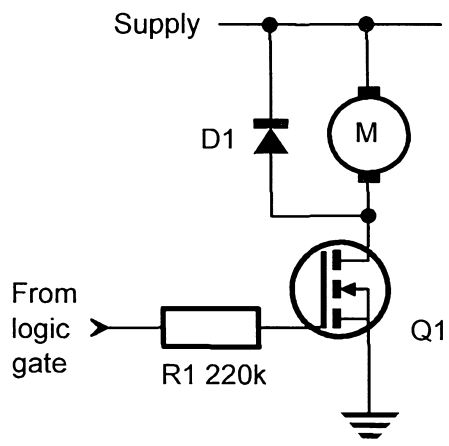


Figure 2

6. What is the name of the diode *D1*?

_____

7. Why is *D1* necessary?

_____

_____

_____

8. What would happen if the diode was accidentally put into the circuit the other way around?

_____

_____

9. What is the advantage of using a MOSFET in a buffer/driver circuit instead of a BJT?

_____

_____

_____

Copyright: Not to be reproduced without permission

10. List the disadvantages of relays compared to optocouplers.

11. What is *switch bounce*?

12. Give an example of a problem that switch bounce might cause.

13. Describe the difference between gates with and without input hysteresis.

14. What is debouncing?

15. What is the main purpose for using an optocoupler at the input/output of a digital circuit?

16. What are the advantages of optocouplers over relays?

Copyright: Not to be reproduced without permission

Student notes

Section 10 RS flip flops

Purpose To develop your understanding of the operation of RS flip flops and introduce you to sequential logic devices and triggering principles.

Objectives At the end of this section you should be able to:

- Recognise the circuit and the schematic symbol of NOR and NAND RS flip flops
- Write the truth table for NOR and NAND RS flip flops
- Give applications where the RS flip flop is used as a latch
- Draw a timing diagram for the NOR and NAND RS flip flops given a set of logic levels for their inputs that change over time
- Explain the difference between *combinational logic* and *sequential logic*
- Explain the difference between *active high* and *active low* combinational and sequential logic inputs
- Explain the difference between *asynchronous* and *synchronous* inputs
- Use a logic probe to test and measure logic sequential logic circuits

Introduction

The treatment of digital electronics so far has concentrated on combinational logic circuits that implement Boolean expressions. The remainder of this subject introduces you to logic devices that implement discrete logic functions (such as counters and shift registers). The building blocks of these more sophisticated devices is the flip flop. This section and the next two introduce you to flip flops and a branch of digital electronics called *sequential logic*.

RS flip flops

The RS flip flop is made from either two cross-coupled NOR gates as shown in Figure 1a below or two cross-coupled NAND gates as shown in Figure 1b.

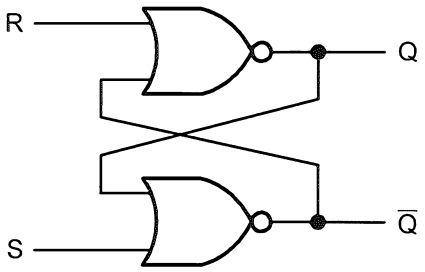


Figure 1a *NOR RS Flip Flop*

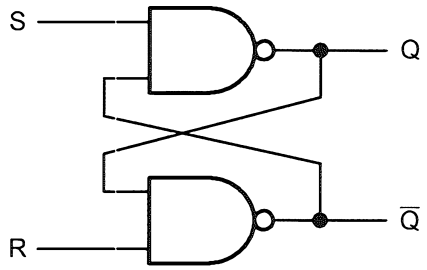


Figure 1b *NAND RS Flip Flop*

Before we examine the operation of these circuits, there are a couple of interesting things to note. First, the outputs are denoted by the letter Q instead of the letter Z and this is common practice for RS flip flops and many sequential logic devices.

Second, both of these flip flops have *complementary* outputs. That is, there is a Q-output and there is a NOT-Q-output ($\overline{Q}$). This labelling tells us that we can expect these two outputs always to be the opposite logic level to each other.

Third, the inputs are labelled using the letters R and S instead of A and B as you have seen on combination logic circuits. These letters are used because when the S-input is *asserted* (that is, activated), the Q-output is *set* and when the R-input is asserted, the Q-output is *reset*. (Recall from your earlier work "set" and "reset" are alternate words for logic-1 and logic-0.) Clearly then, the letters S and R are used as an abbreviation for the words *set* and *reset*.

The basic operation of the NOR and NAND RS flip flops is the same but there are some small differences between so we'll examine their operation separately starting with the NOR RS flip flop. Here on, the term flip flop is usually abbreviated to FF.

NOR RS FF operation

As the RS FF has two inputs, there are four possible combinations of logic levels that can be applied to them. They are: 00, 01, 10 and 11. The circuits in Figure 2 show what happens to the outputs of a NOR RS FF when 01 and 10 are applied to the inputs. [Note: You can work this out for yourself but, to do so, you must assume the Q and $\overline{Q}$ outputs are already some logic level. It doesn't matter which logic level you choose.]

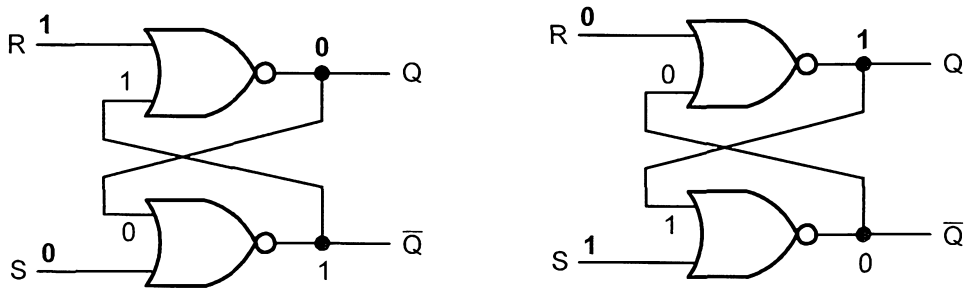


Figure 2 *The outputs of a NOR RS FF when R = 1 & S = 0 and when R = 0 & S = 1*

As you can see, if the R-input is a logic-1 and the S-input is a logic-0, the Q-output is reset (that is, logic-0). And, if the S-input is a logic-1 and the R-input is a logic-0, the Q-output is set (that is, logic-1).

Figure 3 below shows that, if 00 is applied to the inputs, the outputs don't change. That is, if the Q-output is reset then it stays reset and if the Q-output is set then is stays set.

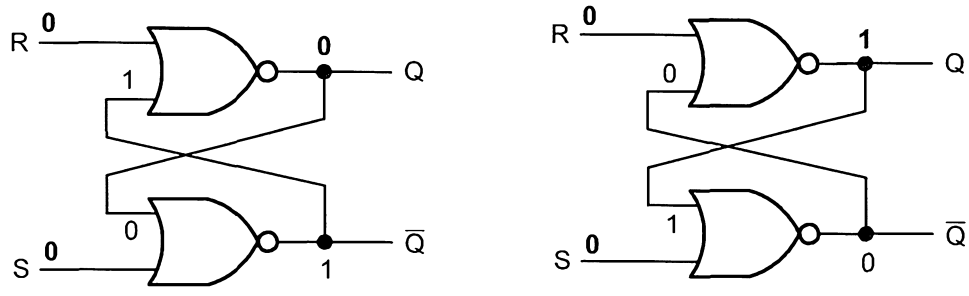


Figure 3 *The output of a NOR RS FF doesn't change if both of the inputs are logic-0*

The fact that a logic-0 on both inputs of a NOR RS FF has no effect on the flip flop's output is significant. It tells us that the logic level that asserts an input is a logic-1. If you look back at the circuits in Figure 2 you'll see that this is demonstrated. Notice that when the R-input is a logic-1, the Q-output is reset (though the S-input must remain a logic-0 for this to happen). And, when the S-input is a logic-1, the Q-output is set (though the R-input must remain a logic-0 for this to happen). Consequently, the inputs to a NOR RS FF are said to be *active high.*

As the *R* and *S* inputs are active high, a problem arises when 11 is applied to the inputs (that is, both inputs are asserted at the same time). Based on the discussion so far, 11 would be trying to simultaneously set and reset the Q-output. However, as you know, it is impossible for a logic gate's output to be both a logic-1 and a logic-0 at the same time. Figure 4 below shows what actually happens when both inputs are asserted at the same time.

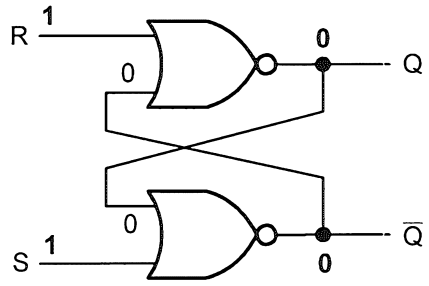


Figure 4 *The logic levels around the circuit of a NOR RS FF when both inputs are logic-1*

As you can see, the Q-output is reset but so is the $\overline{Q}$-output. Obviously, this is a problem because, by definition, the $\overline{Q}$-output is supposed to be the opposite of the Q-output. In terms of the labelling of the circuit's outputs, the circuit's operation has broken-down and, without additional circuitry, there is no way to prevent this. So, we say that 11 on the input to a NOR RS FF is not allowed.

The NOR RS FF's operation can be summarised using a truth table. This is shown in Table 1 below.

Table 1

S	R	Q	$\overline{Q}$	Description
0	0	Q	$\overline{Q}$	No change
0	1	0	1	Q is reset
1	0	1	0	Q is set
1	1	0	0	Disallowed

Instead of drawing the whole circuit, the NOR RS FF can be represented by the schematic symbol shown in Figure 5 below. You may sometimes see the same symbol drawn but with the S and R inputs swapped around and/or the Q and $\overline{Q}$ outputs swapped around. This is usually done for drawing convenience and doesn't imply that the circuit is in any way different.

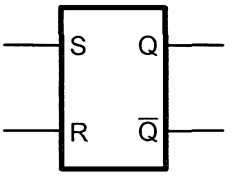


Figure 5 *The schematic symbol of a NOR RS FF*

NAND RS FF operation

The circuits in Figure 6 show what happens to the outputs of a NAND RS FF when 01 and 10 are applied to the inputs.

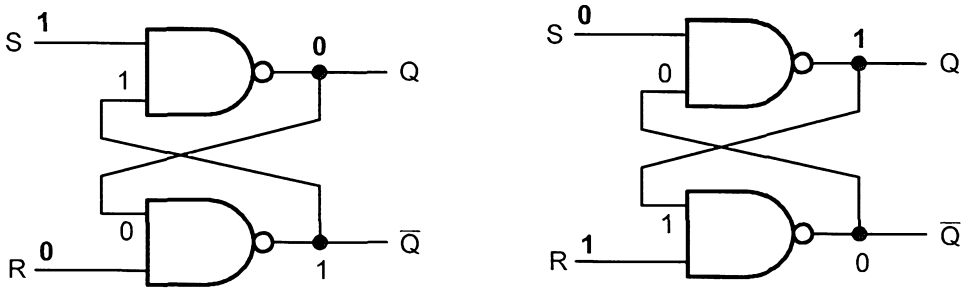


Figure 6 *The outputs of a NAND RS FF when R = 0 and S = 1 and when R = 1 and S = 0*

As you can see, if the R-input is a logic-0 and the S-input is a logic-1, the Q-output is reset. And, if the S-input is a logic-0 and the R-input is a logic-1, the Q-output is set.

Figure 7 shows that, if 11 is applied to the inputs, the outputs don't change.

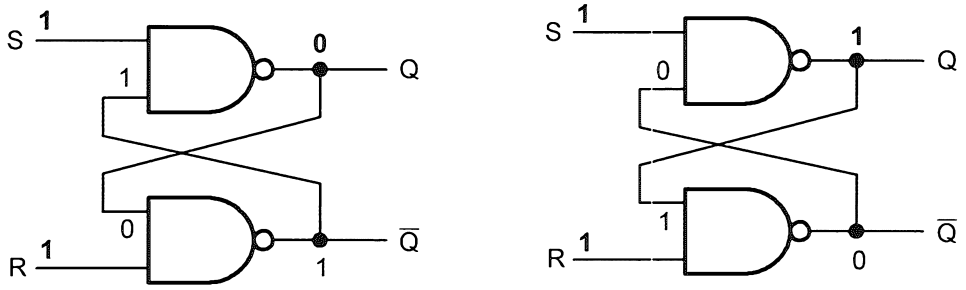


Figure 7 *The output of a NAND RS FF doesn't change if both of the inputs are logic-1*

From the circuits in Figures 6 and 7 we can see that the logic level that asserts the inputs is a logic-0. In other words, the inputs are said to be *active low*.

Figure 8 shows the logic levels around the circuit of a NAND RS FF when both inputs are a logic-0. As you can see, the Q-output is set but so is the $\overline{Q}$-output. So, we say that asserting both inputs to a NAND RS FF at the same time is not allowed.

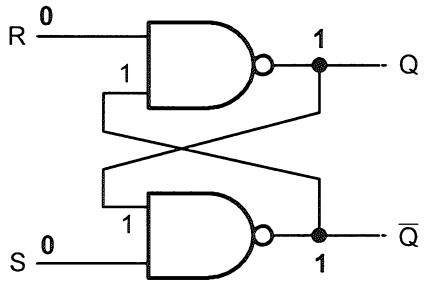


Figure 8 *The logic levels around the circuit of a NAND RS FF when both inputs are logic-0*

The NAND RS FF's operation is summarised in Table 2 below. Notice that the NOT bar is written above the letters S and R to denote that they are active low inputs.

Table 2

$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$	Description
0	0	1	1	Disallowed
0	1	1	0	Q is set
1	0	0	1	Q is reset
1	1	Q	$\overline{Q}$	No change

Instead of drawing the whole circuit, the NAND RS FF can be represented by the either one of the schematic symbol shown in Figure 9 below. Notice that the left-side symbol uses little circles to denote that inputs are active low. The right-side symbol uses lines above S and R instead. As you will see, these two notations are used widely in digital electronics.

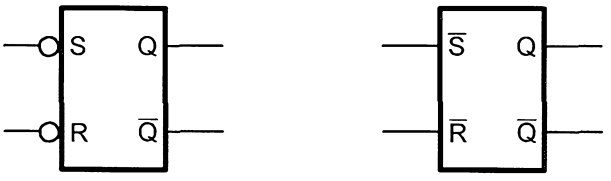


Figure 9 *The schematic symbol of a NAND RS FF*

The RS flip flop as a latch

There are many applications for the RS FF but one of the most useful is the *latch*. The word "latch" in everyday English means to hold something and stop it from moving or changing position (for example, to latch a gate means that you've stopped it from swinging). In digital electronics it is often necessary to capture a logic level and not allow it to change back. For example, consider the simple alarm system shown in Figure 10 below.

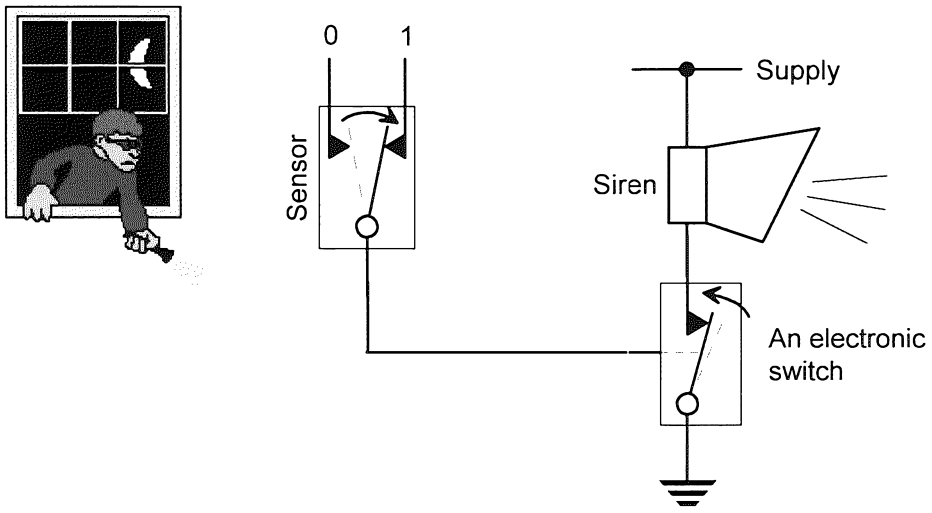


Figure 10 *The conditions around the circuit of an alarm when the sensor detects a person's presence*

When the sensor detects a person's presence, its switch is activated sending a logic-1 to the control input of the electronic switch causing it to close and the siren to sound. However, if the sensor is a pressure mat or a movement detector then, when the person steps off the mat or stops moving, the sensor's switch returns to the default position sending a logic-0 to the control input of the electronic switch causing it to open and the siren to turn off.

This circuit would be suitable for an alarm at the door of a shop to indicate that a customer has walked in but it is not suitable for a building alarm for detecting a break-in. Building alarms are required to continue to sound even if the sensor no-longer detects the intruder's presence. In other words, once the alarm has been tripped, it needs to stay tripped until it is manually reset.

The simple circuit in Figure 10 can be modified to make it more useable as a building alarm with the addition of an RS FF used as a latch. This is shown in Figure 11 below.

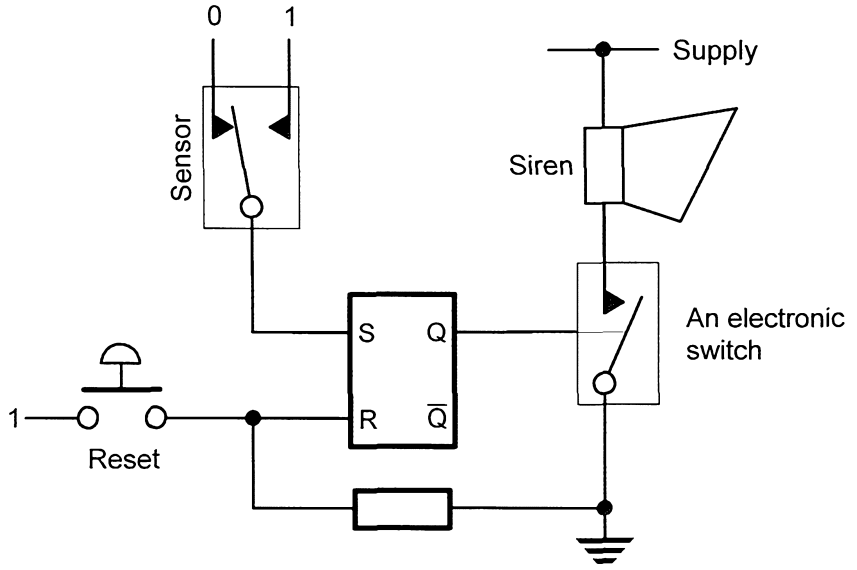


Figure 11 *The circuit in Figure 10 with the addition of a NOR RS FF*

Assuming the alarm has been reset, if the sensor doesn't detect an intruder the logic levels around the circuit are as shown in Figure 12 below.

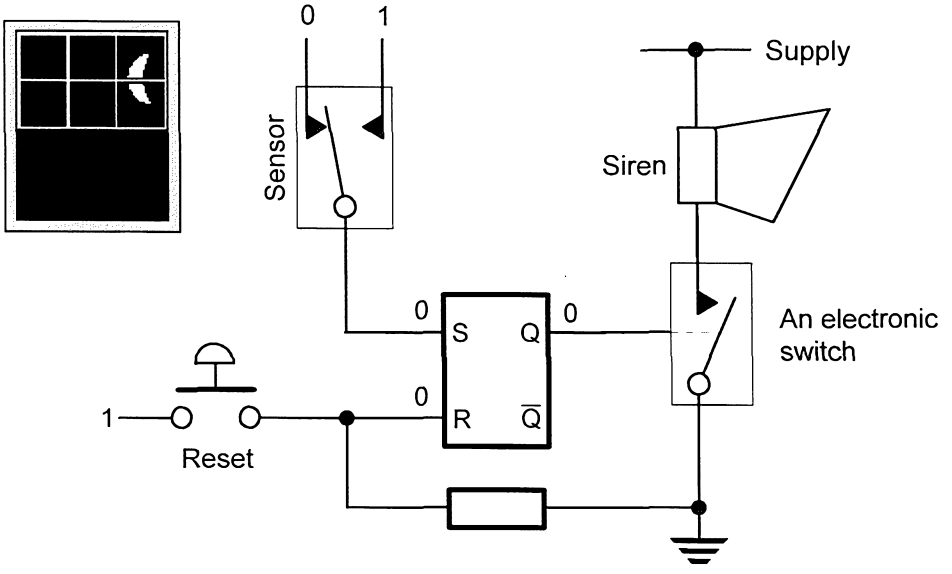


Figure 12 *The conditions around the circuit when there is no intruder*

Then when the sensor detects an intruder, its switch connects a logic-1 to the flip flop's S-input. As we know, when the S-input of a NOR RS FF is logic-1 and the R-input is logic-0, the Q-output gets set and this is shown in Figure 13 below. This closes the electronic switch and the siren sounds.

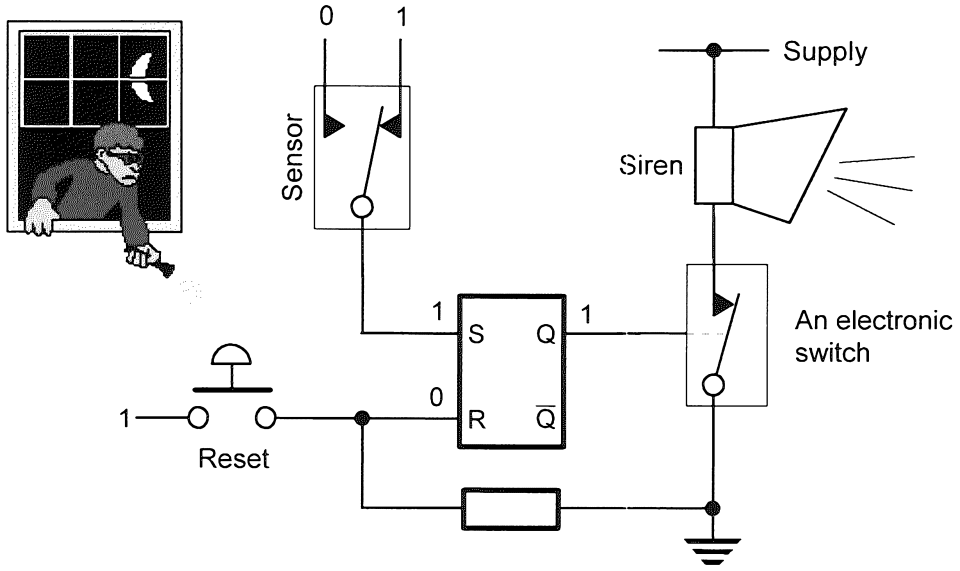


Figure 13 *The conditions around the circuit when the sensor detects an intruder*

If the intruder is scared-off and the sensor no-longer detects their presence, the sensor's switch reconnects a logic-0 to the flip flop's S-input. But, a logic-0 on both inputs to a NOR RS FF doesn't cause a change on its Q-output. This means that, once the alarm has been tripped it stays tripped. This is shown in Figure 14 below.

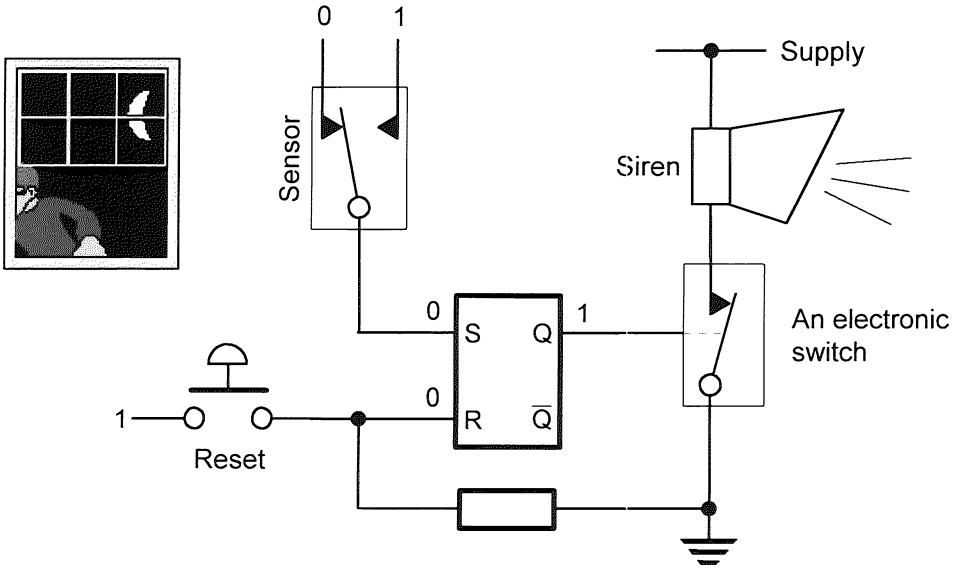


Figure 14 *The conditions around the circuit when the sensor no-longer detects an intruder*

To reset the alarm, the reset switch must be pressed. This momentarily asserts the NOR RS FF's R-input resetting the Q-output, opening the electronic switch and turning off the siren. This is shown in Figure 15 below.

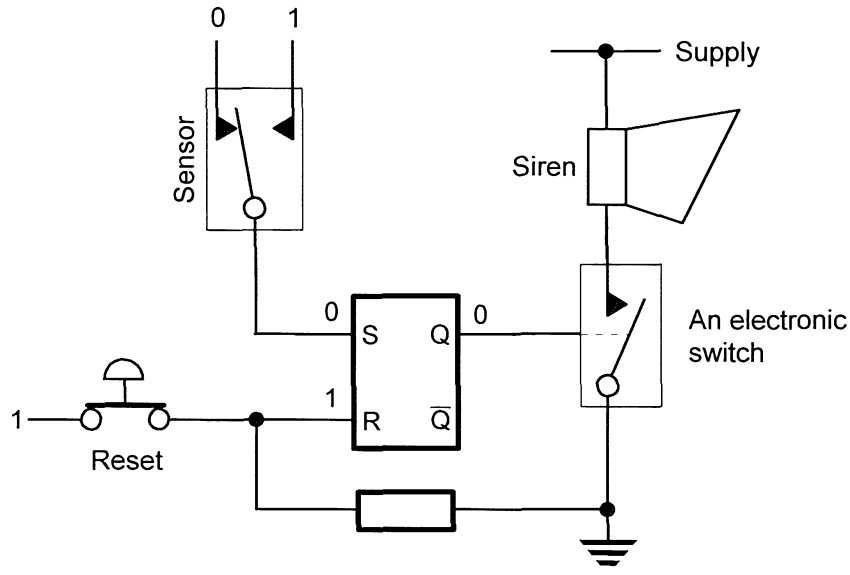


Figure 15 *The conditions around the circuit when the reset button is pressed*

There are many examples of this type of switch-latching in digital electronics including: the call indicators for situations like elevators in buildings, nurses in hospitals and flight attendants on planes; the memory LED on a logic probe, and the pedestrian crossing button on a set of traffic lights.

Another application for the flip flop is as a storage for a single bit of data. As long as the circuit is powered, a flip flop can store a logic-0 or a logic-1 indefinitely. Groups of flip flops connected together can provide significant amounts of storage for digital information and, as may learn in an elective unit, this is the basis of *SRAM* memory in personal computers.

Timing diagrams

Electronics repair technicians are sometimes interested in what happens to the output of a sequential logic circuit over time. Moreover, the operation of some sequential logic circuits can only be understood in the first instance when we look at what happens over time. To make this easier to do, the information is represented visually using a *timing diagram*.

A timing diagram is simply a graph of the input and output logic levels of a sequential logic circuit versus time. An example is shown in Figure 16 below showing what happens on the Q and $\overline{Q}$ outputs as the logic levels on the R and S inputs of a NOR RS FF change. Obviously, if the inputs are different logic levels at different times the shape of the signal on the Q and $\overline{Q}$ outputs will be different.

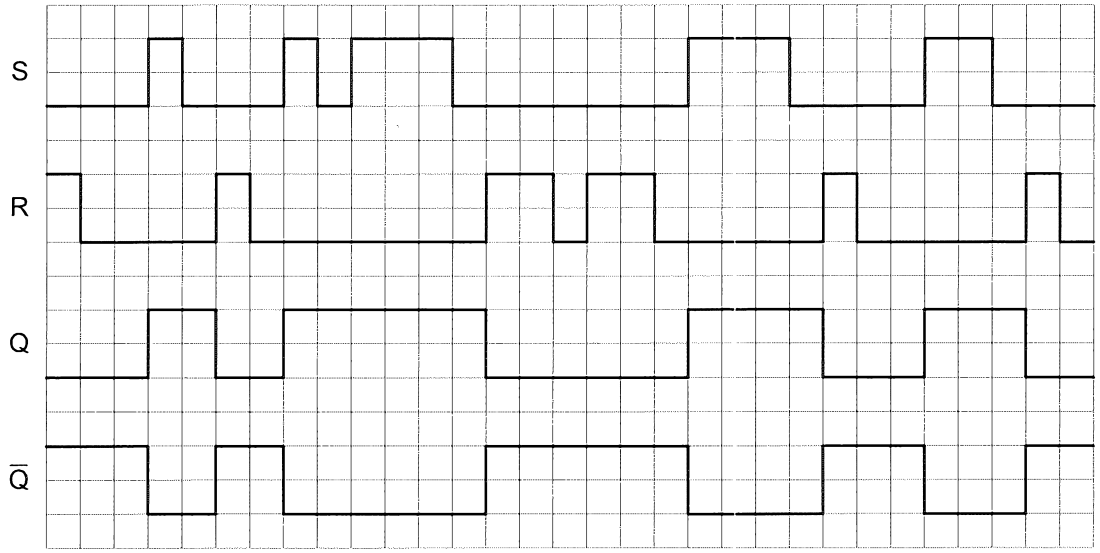


Figure 16 *The timing diagram for a NOR RS FF for the inputs given*

You're expected to be able to draw timing diagrams for NOR and NAND RS flip flops given a set of logic levels on the inputs. The procedure for this follows.

Drawing a timing diagram

To help explain how to draw a timing diagram for RS flip flops, the procedure is demonstrated by showing how the timing diagram in Figure 16 (on the previous page) was drawn.

1. Write down the truth table of the flip flop being analysed in free space close to the timing diagram.

   The timing diagram in Figure 16 is for a NOR RS FF so let's write its truth table.

S	R	Q	$\overline{Q}$	Description
0	0	Q	$\overline{Q}$	No change
0	1	0	1	Q is reset
1	0	1	0	Q is set
1	1	0	0	Disallowed

2. Starting from the most left-hand side of the graph, identify the first period of time over which the logic levels don't change on the *R* and *S* inputs. See Figure 17.

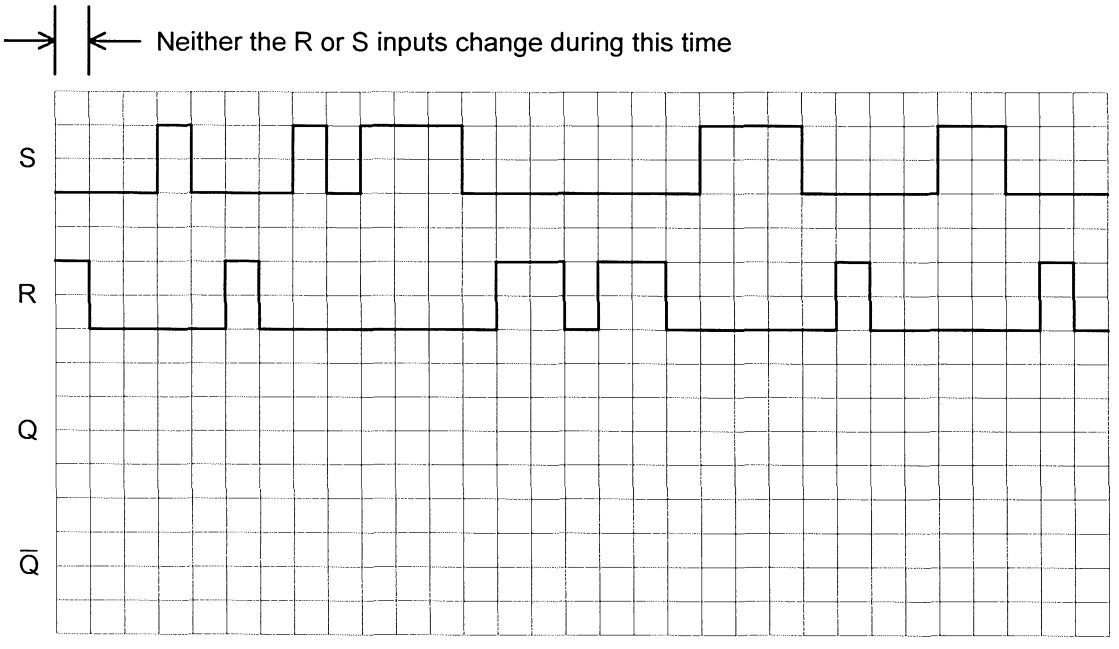


Figure 17

3. Write down the logic levels on the inputs during this time. See Figure 18.

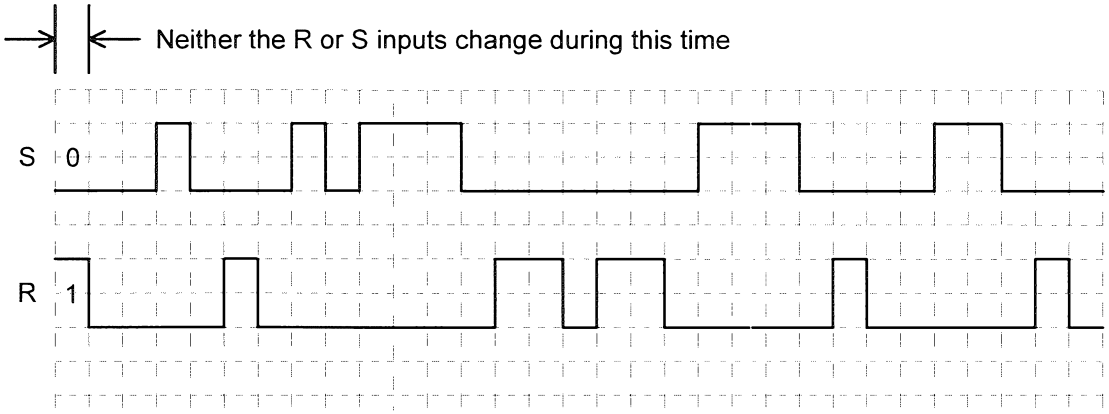


Figure 18

4. Use the truth table to determine the logic level on the Q-output for the input conditions identified.

S	R	Q	$\overline{Q}$	Description
0	0	Q	$\overline{Q}$	No change
0	1	0	1	Q is reset
1	0	1	0	Q is set
1	1	0	0	Disallowed

5. Draw a line for the logic level that the Q-output should during that time. See Figure 19.

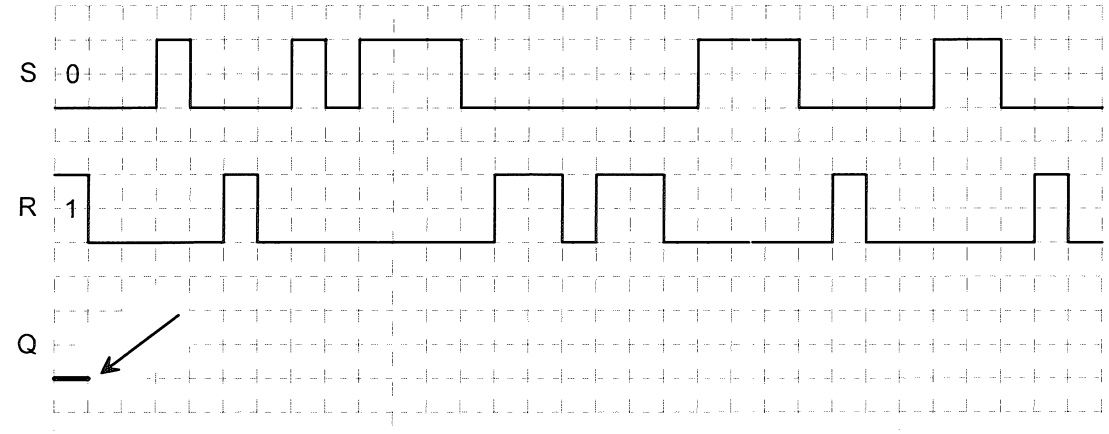


Figure 19

6. Identify the next period of time over which the logic levels don't change on the *R* nor *S* inputs change and repeat. See Figures 20 to 23.

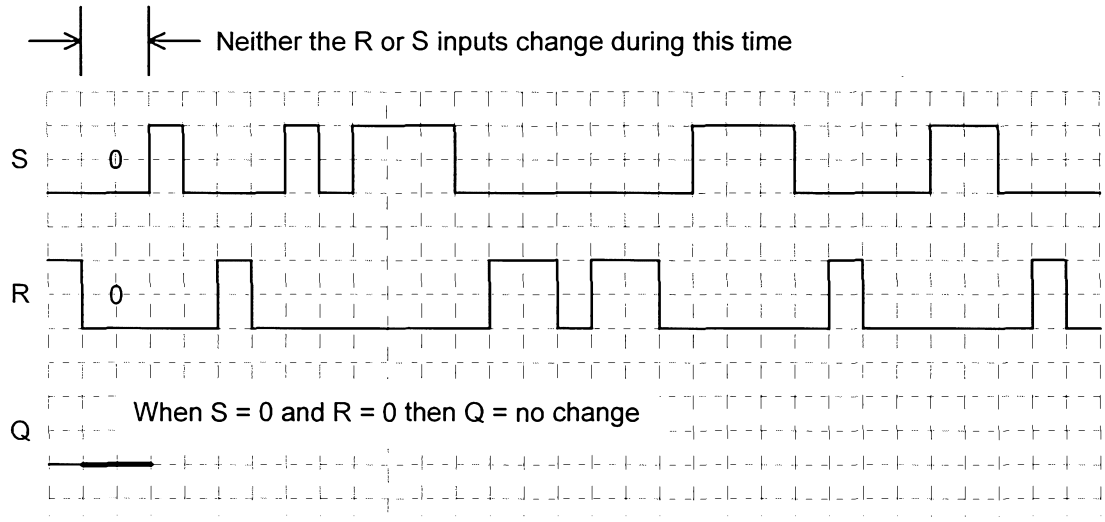


Figure 20

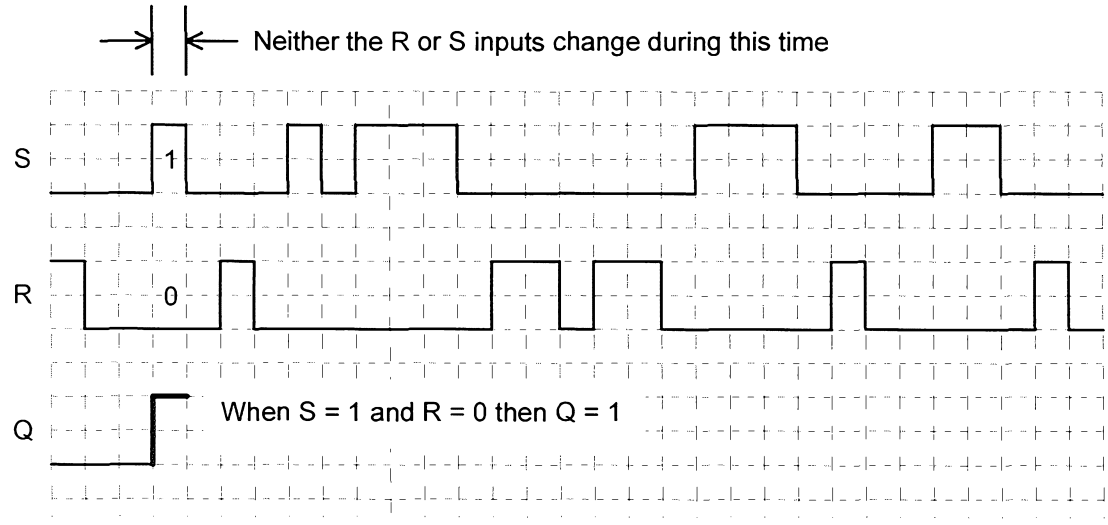


Figure 21

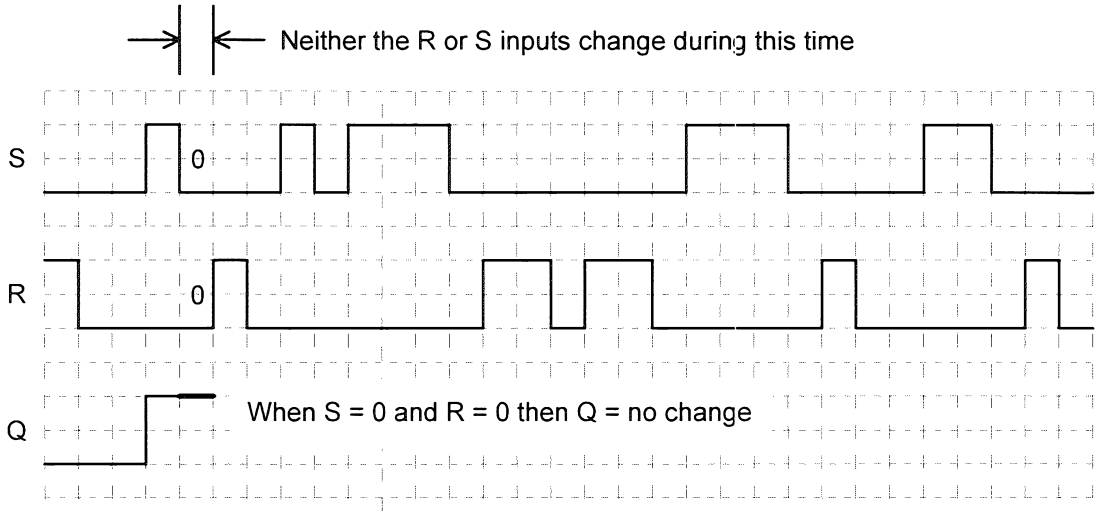


Figure 22

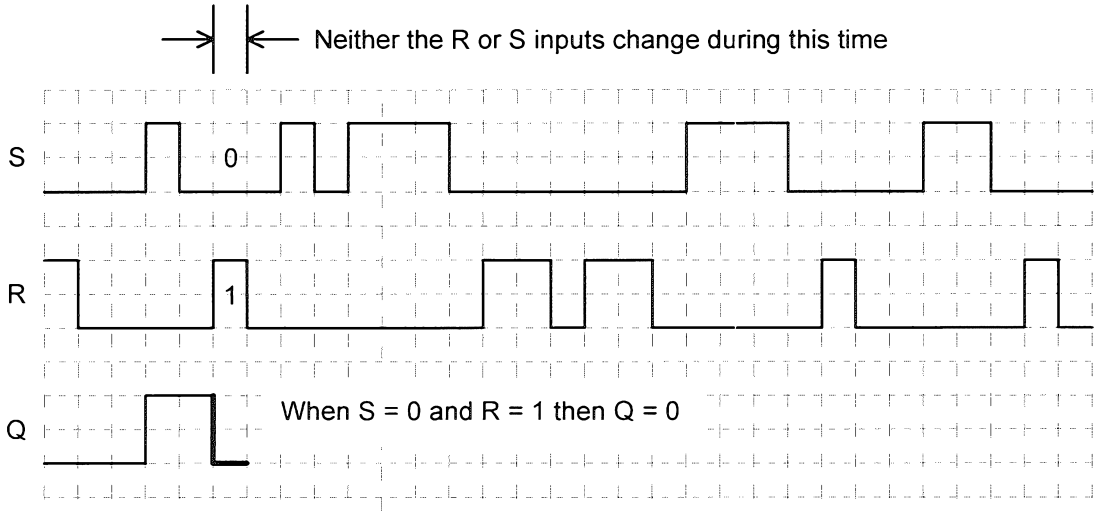
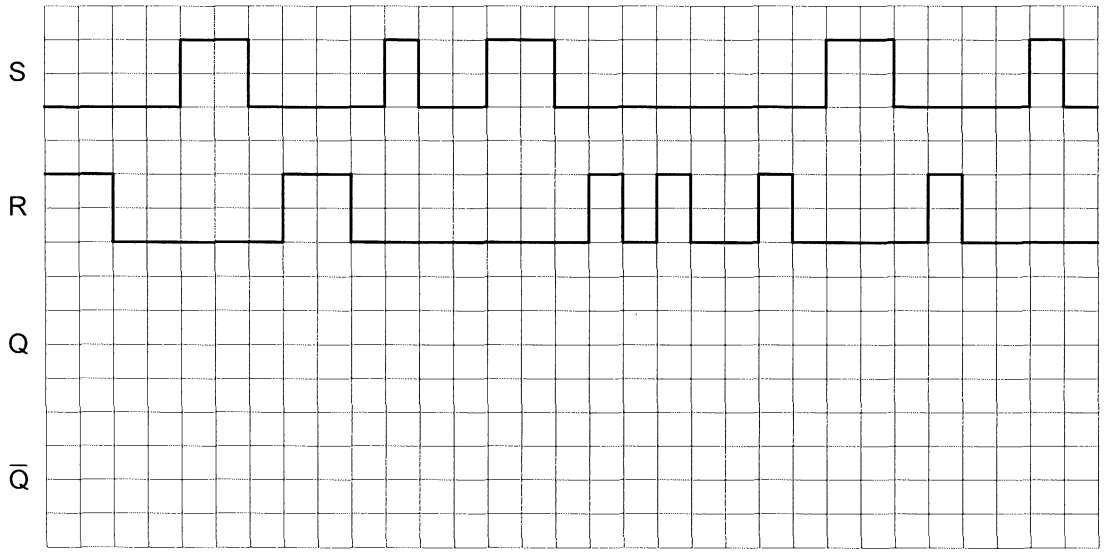


Figure 23

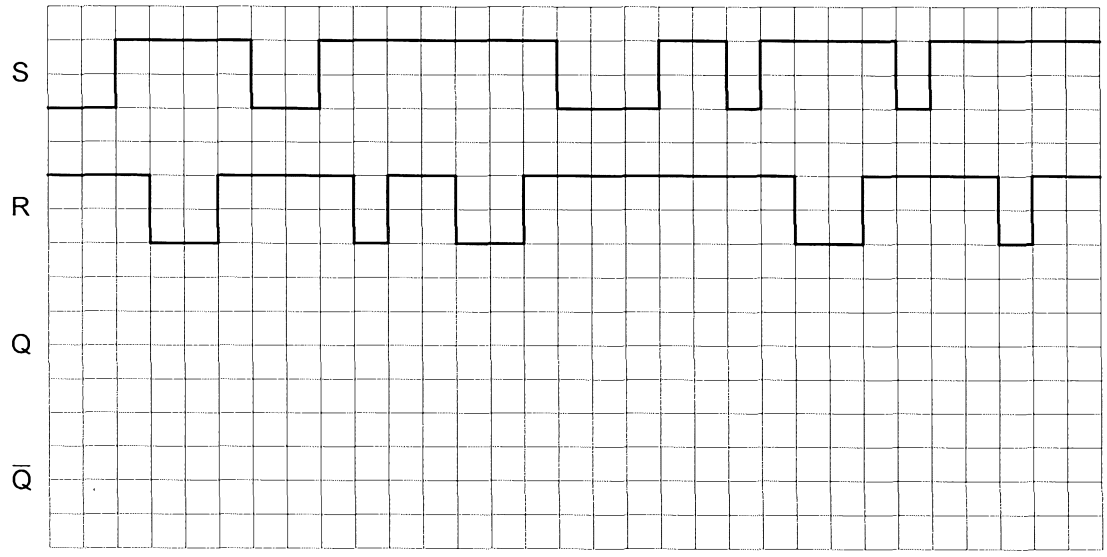
7. Once all the logic levels for the Q-output are drawn, draw the logic levels for the $\overline{Q}$-output by simply drawing the Q-output upside down. This can be seen in Figure 16 (on page 10-12).

Practise drawing timing diagrams for the outputs of RS flip flops for yourself by trying the following questions.

1. Complete the timing diagram below for a NOR RS FF.



2. Complete the timing diagram below for a NAND RS FF.



Combinational versus sequential logic

By introducing you to flip flops, your study of digital electronics has evolved in an important way. As you know, combinational logic circuits have outputs that are determined by the logic level that is presently on their inputs. However, flip flops are different. Their outputs are determined by the logic level presently on their inputs and by the logic level on their inputs previous to their present values.

If this sounds ridiculous, consider the NOR RS flip flop in Figure 24 below. It presently has the following conditions, $R = 1$, $S = 0$. Consequently, the flip flop's present outputs are $Q = 0$ and $\overline{Q} = 1$.

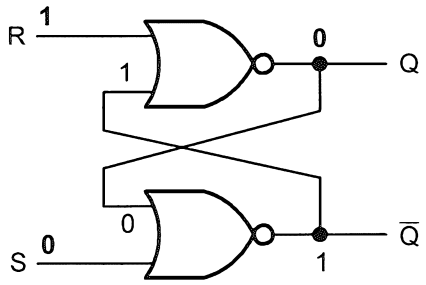


Figure 24

Importantly, the outputs are fed back to the opposing logic gates' unlabelled input and must play a part in determining what will happen to the logic gates' outputs when R and S next change. That being the case, Q and $\overline{Q}$ are affected by the logic levels on R and S both before they change and after.

Putting this another way, flip flop outputs are a function of the history of their inputs. As such, flip flops are classed as a different type of digital electronics called *sequential logic* (because the sequence of logic levels that have been applied to the inputs previously affect their present condition). And, as flip flops form the basis of discrete logic devices such as and *shift registers*, *counters*, etc these devices are also considered sequential logic.

Asynchronous versus synchronous inputs

The NOR and NAND RS flip flops discussed so far are examples of sequential logic devices with inputs that are said to be *asynchronous*. This is because the inputs are continuously read. Importantly, they're not synchronised to operate in time with any other signal.

In complex digital electronics equipment, things must happen in the correct sequence and at the correct time for the equipment to work properly. For example, there is no point in a PVR attempting to read the data on the hard disc drive if the disc isn't spinning. This means that electrical events around complex digital circuits cannot be allowed to happen in their own time. They must occur in the right order and at the right time. For this reason, control is exercised over when some inputs to sequential logic devices are read by making them *synchronous*. Synchronous inputs are not read by the device until a "trigger" signal is applied to an extra input.

Triggering principles

Circuit design provides for two different methods of the triggering the synchronous inputs of sequential logic devices. One discussed here allows the inputs to be read for the whole time that the trigger input is a particular logic level and this method is said to be *level-sensitive*. Level-sensitive triggering inputs are usually called the *gate* (*G*) or the *enable* (*E*).

A positive level-sensitive gate/enable input on a sequential logic device allows it to read its synchronous inputs during the time that the gate is a logic-1 (see Figure 25a). It is common in digital electronics to say that this type of gate input is "active high". A negative level-sensitive gate/enable input allows the device to read its synchronous inputs while the gate signal is a logic-0 (see Figure 25b). In this case, the gate input is said to be "active low".

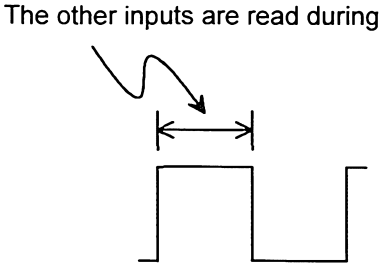


Figure 25a *Active high gate/enable*

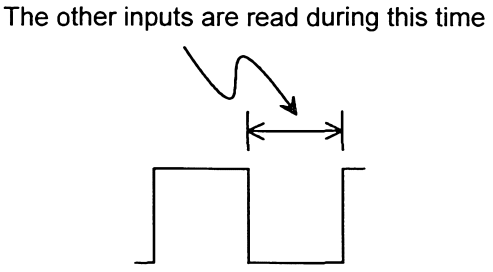


Figure 25b *Active low gate/enable*

A gate-controlled RS flip flop

To explain the relationship between the gate input of a sequential logic RS FF and its other inputs, a NOR RS FF with an active high gate input is used. Figure 26 shows the device's schematic symbol with its truth table next to it in Table 3. [Note: The actual circuit of a NOR RS FF with an active high gate input is not shown here because you're not expected to be able to draw it or explain its operation. However, the circuit and its operation are provided in Appendix 2 for your information if you're interested.]

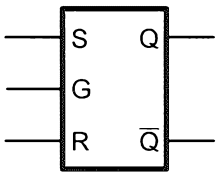


Figure 26

Table 3

G	S	R	Q	$\overline{Q}$	Description
0	X	X	Q	$\overline{Q}$	Inputs not read, no change
1	0	0	Q	$\overline{Q}$	Inputs read but no change
1	0	1	0	1	Q is reset
1	1	0	1	0	Q is set
1	1	1	0	0	Disallowed

The first row of the truth table shows that when the gate is a logic-0, it doesn't matter what logic level is on the *S* and *R* inputs (hence the use of the letter *X*), the *Q* and $\overline{Q}$ outputs remain unchanged. In other words, the device cannot read its inputs. But, as soon as the gate input is a logic-1 the device works in exactly the same way as a normal RS FF.

The operation of this type of flip flop might be better understood if demonstrated using a timing diagram. Figure 27 on the next page shows the effect on the *Q* and $\overline{Q}$ outputs for the *S* and *R* inputs signals given. For the time that the gate is a logic-1, the circuit works exactly like a normal NOR RS FF. But notice that, once the gate input is made logic-0, the *Q* and $\overline{Q}$ outputs are frozen at the last logic level before the device was disabled - changes on *S* and *R* are ignored.

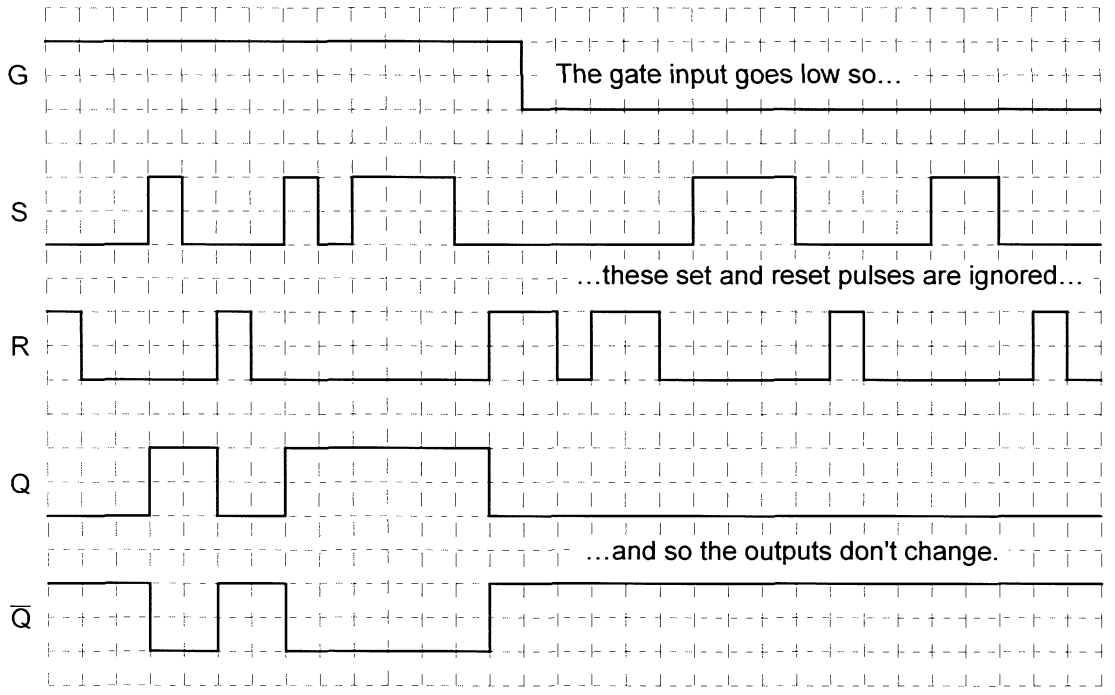


Figure 27

A minor modification to the circuit (explained in Appendix 2) changes the gate from active high to active low. Obviously the schematic symbol must be changed to denote this difference and this is shown in Figure 28 below. The device's truth table is shown in Table 4.

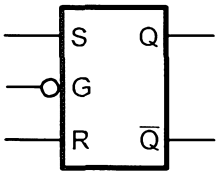


Figure 28

Table 4

$\overline{G}$	S	R	Q	$\overline{Q}$	Description
1	X	X	Q	$\overline{Q}$	Inputs not read, no change
0	0	0	Q	$\overline{Q}$	Inputs read but no change
0	0	1	0	1	Q is reset
0	1	0	1	0	Q is set
0	1	1	0	0	Disallowed

Although they're not shown here, NAND RS FFs can also be made with active high and active low gate/enable inputs.

Words that may be new to you

Active low input	Any input that is asserted for the whole time that it is a logic-0.
Active high input	Any input that is asserted for the whole time that it is a logic-1.
Asynchronous input	Any input (other than a gate, enable or clock) that the device reads immediately.
Clock input	A type of trigger input that is usually edge-sensitive.
Combinational logic	A type of logic circuit or device that only has asynchronous inputs.
Gate input	A trigger input that is usually level-sensitive.
Enable input	A trigger input that is usually level-sensitive.
Level-sensitive input	Any input that is asserted for the whole time the input is a logic-0 or a logic-1.
Sequential logic	A type of logic circuit or device whose outputs are a function of its present and previous inputs.
Synchronous input	Any input (other than a gate, enable or clock) that the device only reads when it is told to via a trigger input such as a gate, enable or clock.

Skill practice 10

Practise testing and measuring sequential logic circuits (RS flip flops)

This exercise is practise for the sorts of skills you may be required to perform in a practical test. Remember, in any practical tests you will be working alone so make sure that you can perform all the steps. It should take you about 1¼ hours to complete this exercise.

Equipment

- Digital trainer
- logic probe
- 74LS00
- 74LS02
- 74LS08
- hook-up wire (terminated)
- IC extractor

Remember:

Follow TAFE NSW WHS guidance at all times!

 Copyright: Not to be reproduced without permission

Work tasks

1. Read your WHS responsibilities at the top of the form below. Then conduct a WHS risk assessment and record your findings in the space provided.

Responsibilities of students under the Model WHS Act: s28

- Take reasonable care for your own health and safety by working safely at all times
- Take reasonable care to ensure that your acts or omissions don't put the health and safety of others at risk
- Follow all TAFE NSW WHS guidance and comply with all reasonable instructions from TAFE NSW staff to assist them in complying with the TAFE NSW WHS requirements
- In addition to the above, you must:
  - use and maintain machinery, tools and all other equipment properly and safely
  - ensure that your work area is free of hazards
  - notify a TAFE NSW staff member of actual or potential hazards
  - wear/use prescribed safety equipment
  - take notice of any safety signs and adhere to their instructions

Risks involved in this activity include:	**Control measures:**
■ Trip hazards (eg students bags) ■ Objects dropped on feet (while equipment is taken to and from benches).	■ Move bags and other objects from walkways ■ Plan lifting of equipment
Others: _____ _____ _____	Other: _____ _____ _____

My signature here indicates that I have read and understand my responsibilities under the Model WHS Act s28 (detailed above). I have also conducted a risk assessment before undertaking this activity and have identified measures to control these risks and have implemented them.

Signature: _____ Date: _____

2. Gather the equipment needed for this exercise.

3. Read the following points and make sure that you perform them when wiring all circuits:

   - Write the IC pin numbers of the gates you work with on the circuit that you're instructed to wire.
   - Handle all ICs carefully - remember ESD can damage them.
   - Ensure the power supply is connected to all ICs in the circuit.
   - When plugging hook-up wire into the breadboard to make connections to the IC's pins, use holes that are furthest away from the pin.

4. Wire the circuit of Figure 1 on the digital trainer. Pinout data for the 74LS00 is at the back of the book.

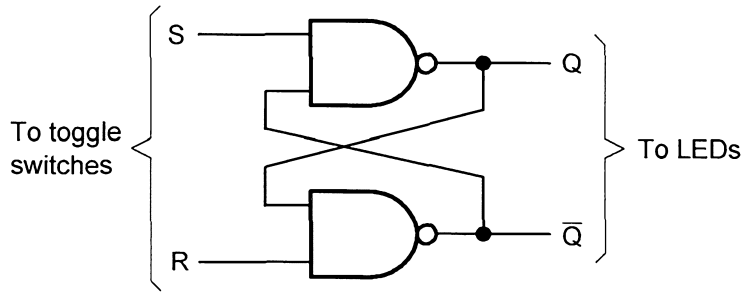


Figure 1

5. Use the toggle switches to make the *R* and *S* inputs logic-1.

6. Switch on the trainer and observe the logic levels on the outputs.

   Self check: The logic level on the two outputs should be the opposite of each other. If they're not, turn off the trainer and check your wiring. If you can't find a wiring problem then carefully swap the IC with another 74LS00. If this doesn't solve the problem, call the teacher.

7. If the **Q-output** is a logic-0 then just jump to the next step. If the Q-output is **not** a logic-0 then turn the trainer on and off a couple of times until it is. If this doesn't work, call the teacher and s/he will make the Q-output go to logic-0 for you.

8. Use the toggle switches to make the S-input a logic-0 and the R-input a logic-1. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

   Q = _____

   $\overline{Q}$ = _____

9. Make the S-input a logic-1 and the R-input a logic-1. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

   Q = _____

   $\overline{Q}$ = _____

10. Make the S-input a logic-1 and the R-input a logic-0. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

    Q = _____

    $\overline{Q}$ = _____

11. Make the S-input a logic-1 and the R-input a logic-1. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

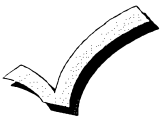
    Q = _____

    $\overline{Q}$ = _____

12. Make the S-input a logic-0 and the R-input a logic-0. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

    Q = _____

    $\overline{Q}$ = _____



The teacher needs to check your work at this point...

Question 1
What is the name of this circuit?

Question 2
Complete Table 1 below from your observations in steps 8 to 12.

Table 1

S	R	Q	$\overline{Q}$	Description
0	0			
0	1			
1	0			
1	1			

Question 3
Are the *S* and *R* inputs active high or active low? Explain your answer.

Question 4
What change should you make to the truth table above to emphasise the fact that the inputs are active low?

The teacher needs to check your work at this point...

13. Wire the circuit of Figure 2 on the digital trainer. Pinout data for the 74LS02 is at the back of the book.

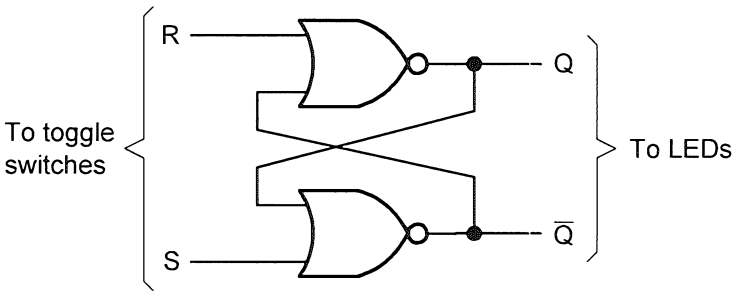


Figure 2

14. Use the toggle switches to make the *R* and *S* inputs logic-0.

15. Switch on the trainer and observe the logic levels on the outputs.

    Self check: The logic level on the two outputs should be the opposite of each other. If they're not, turn off the trainer and check your wiring. If you can't find a wiring problem then carefully swap the IC with another 74LS02. If this doesn't solve the problem, call the teacher.

16. If the **Q-output** is a logic-0 then just jump to the next step. If the Q-output is **not** a logic-0 then turn the trainer on and off a couple of times until it is. If this doesn't work, call the teacher and s/he make the Q-output go to logic-0 for you.

17. Make the S-input a logic-1 and the R-input a logic-0. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

    Q = _____

    $\overline{Q}$ = _____

18. Make the S-input a logic-0 and the R-input a logic-0. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

Q = _____

$\overline{Q}$ = _____

19. Make the S-input a logic-0 and the R-input a logic-1. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

Q = _____

$\overline{Q}$ = _____

20. Make the S-input a logic-0 and the R-input a logic-0. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

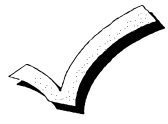
Q = _____

$\overline{Q}$ = _____

21. Make the S-input a logic-1 and the R-input a logic-1. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

Q = _____

$\overline{Q}$ = _____



The teacher needs to check your work at this point...

 Copyright: Not to be reproduced without permission

Question 5
What is the name of this circuit?

_____

Question 6
Complete Table 2 below from your observations in steps 17 to 21.

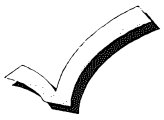
Table 2

S	R	Q	$\overline{Q}$	Description
0	0			
0	1			
1	0			
1	1			

Question 7
Are the *S* and *R* inputs active high or active low? Explain your answer.

_____

_____



The teacher needs to check your work at this point...

22. Wire the circuit of Figure 3 on the digital trainer. Pinout data for the 74LS08 is at the back of the book.

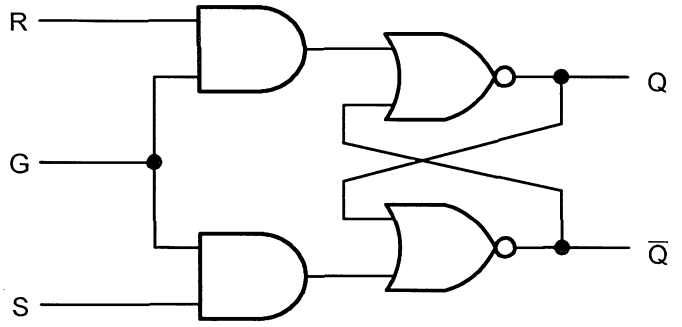


Figure 3

23. Use the toggle switches to make the *R*, *S* and *G* inputs logic-0.

24. Switch on the trainer and observe the logic levels on the outputs.

    Self check: The logic level on the two outputs should be the opposite of each other. If they're not, turn off the trainer and check your wiring. If you can't find a wiring problem then carefully replace the ICs. If this doesn't solve the problem, call the teacher.

25. If the **Q-output** is a logic-0 then just jump to the next step. If the Q-output is **not** a logic-0 then turn the trainer on and off a couple of times until it is. If this doesn't work, call the teacher and s/he will make the Q-output go to logic-0 for you.

26. For steps 27 to 29, keep the G-input a logic-0.

27. Make the S-input a logic-1 and the R-input a logic-0. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

    Q = _____

    $\overline{Q}$ = _____

28. Make the S-input a logic-0 and the R-input a logic-1. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

Q = _____

$\overline{Q}$ = _____

29. Make the S-input a logic-1 and the R-input a logic-1. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

Q = _____

$\overline{Q}$ = _____

30. Make the *R* and *S* inputs logic-0.

31. Make the G-input a logic-1 and keep it that way for steps 32 to 36.

32. Make the S-input a logic-1 and the R-input a logic-0. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

Q = _____

$\overline{Q}$ = _____

33. Make the S-input a logic-0 and the R-input a logic-0. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

Q = _____

$\overline{Q}$ = _____

34. Make the S-input a logic-0 and the R-input a logic-1. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

Q = _____

$\overline{Q}$ = _____

35. Make the S-input a logic-0 and the R-input a logic-0. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

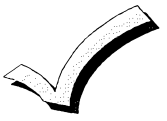
Q = _____

$\overline{Q}$ = _____

36. Make the S-input a logic-1 and the R-input a logic-1. Observe and record what happens (if anything) to the Q and $\overline{Q}$ outputs.

Q = _____

$\overline{Q}$ = _____



The teacher needs to check your work at this point...

Question 8
What is the difference between the operation of this circuit and the circuit in Figure 2?

_____

Question 9
Complete Table 3 below from your observations in steps 27 to 36.

Table 3

G	S	R	Q	$\overline{Q}$	Description
0					
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Question 10
Is the G-input active high or active low? Explain your answer.

Question 11
Is the G-input level-sensitive or edge-sensitive? Explain your answer.

The teacher needs to check your work at this point...

Review questions

Answer these questions to check your understanding of what you have learnt from this chapter. Doing this will also help to prepare you for the assessments.

Questions 1 to 7 refer to the circuit in Figure 1

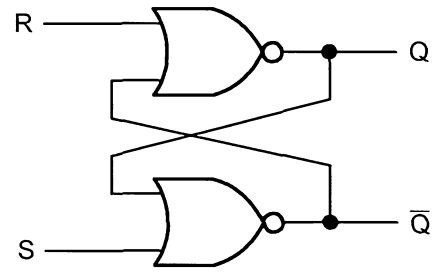


Figure 1

1. What is the circuit's name?

2. What do the letters *R* and *S* stand for?

3. What is the relationship between the circuit's Q and $\overline{Q}$ outputs?

4. Is the circuit an example of combinational logic or sequential logic? Explain your answer.

 Copyright: Not to be reproduced without permission

5. Complete the truth table below for the circuit.

S	R	Q	$\overline{Q}$	Description
0	0	Q	$\overline{Q}$	
0	1			Q is reset
		1	0	Q is set
1	1	0	0	

6. Draw the schematic symbol that can be used to represent the circuit.

7. The inputs to the circuit are

- ☐ asynchronous and active low.
- ☐ synchronous and active low.
- ☐ asynchronous and active high.
- ☐ synchronous and active high.

Questions 8 to 11 refer to the circuit in Figure 2

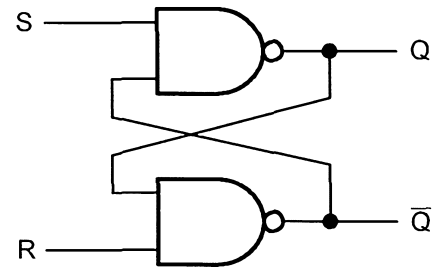


Figure 2

8. What is the circuit's name?

_____

9. Complete the truth table below for the circuit.

$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$	Description
0	0	1	1	
				Q is set
1	0	0	1	
		Q	$\overline{Q}$	No change

10. Draw the schematic symbol that can be used to represent the circuit.

Copyright: Not to be reproduced without permission

11. The inputs to the circuit are

- ☐ asynchronous and active low.
- ☐ synchronous and active low.
- ☐ asynchronous and active high.
- ☐ synchronous and active high.

12. Why are latches sometimes important in digital electronics circuitry?

_____

_____

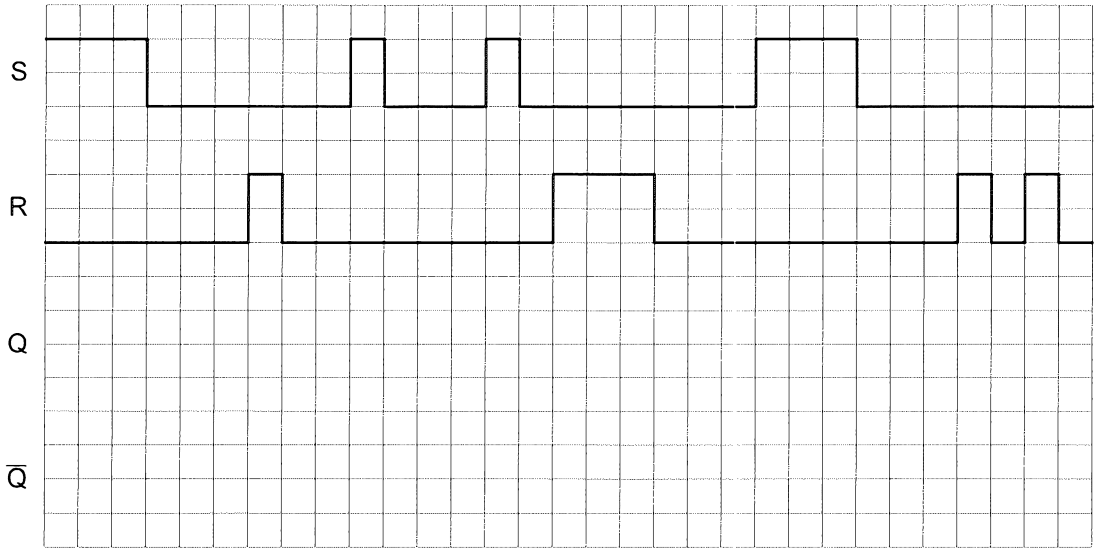
13. Give three examples of where an RS FF can be used as a latch.

_____

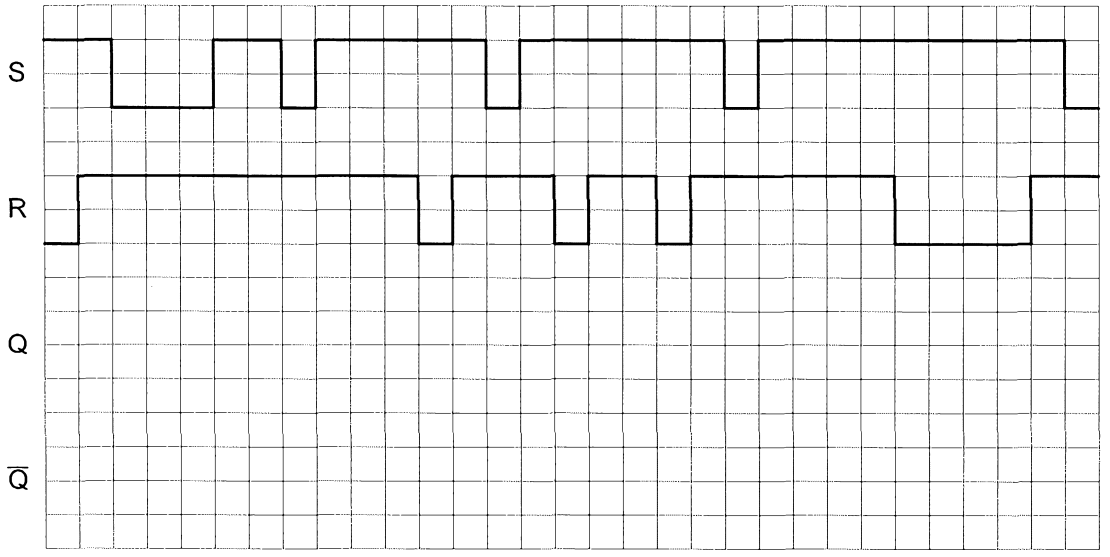
_____

_____

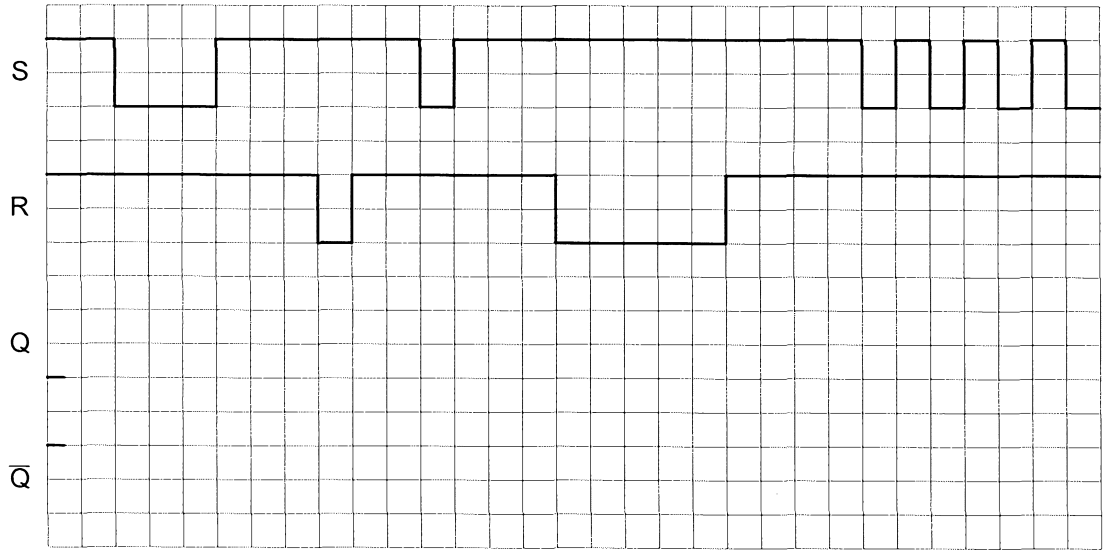
14. Complete the timing diagram below for a NOR RS FF.



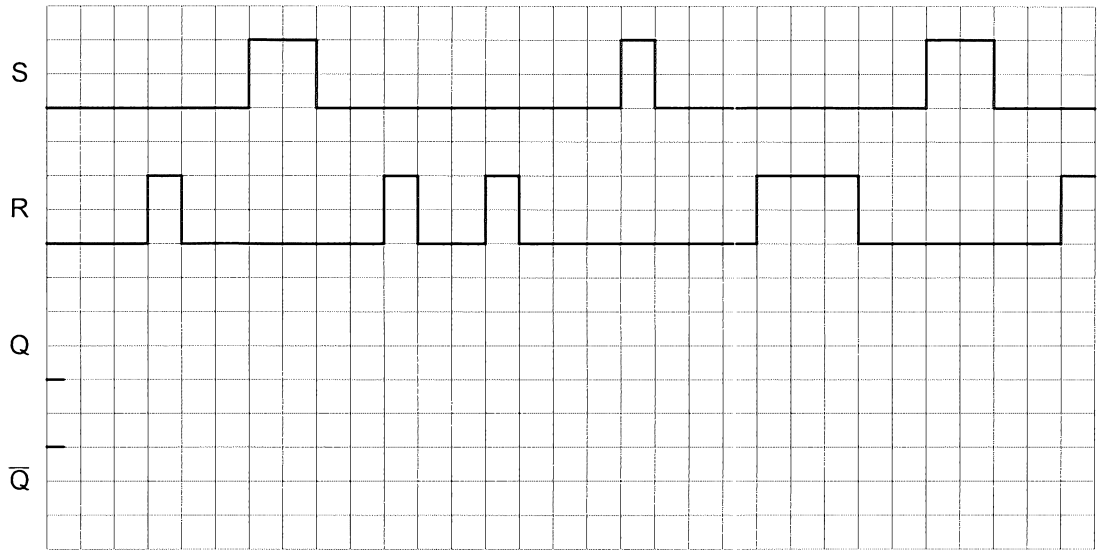
15. Complete the timing diagram below for a NAND RS FF.



16. Complete the timing diagram below for a NAND RS FF. Assume Q starts as logic-0.



17. Complete the timing diagram below for a NOR RS FF. Assume Q starts as logic-0.



Questions 18 and 19 refer to the device in Figure 3 below

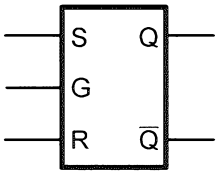


Figure 3

18. Which input or inputs are synchronous?

- ☐ G only
- ☐ R and S only
- ☐ G, R and S
- ☐ None of them

19. Which input or inputs are active high?

- ☐ *G* only
- ☐ *R* and *S* only
- ☐ *G*, *R* and *S*
- ☐ None of them

Questions 20 and 21 refer to the device in Figure 4 below

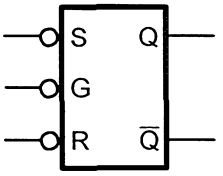


Figure 4

20. Which input or inputs are asynchronous?

- ☐ *G* only
- ☐ *R* and *S* only
- ☐ *G*, *R* and *S*
- ☐ None of them

21. Which input or inputs are active low?

- ☐ *G* only
- ☐ *R* and *S* only
- ☐ *G*, *R* and *S*
- ☐ None of them

 Copyright: Not to be reproduced without permission

Student notes

Appendix 1

 Copyright: Not to be reproduced without permission

Digital test equipment

In the electronics subjects that you have studied so far, you have learnt how to use a multimeter to measure voltages around circuits. This activity is a common way that technicians obtain information about whether or not a circuit is working. The measured voltages are compared to the theoretically expected voltages and if they are similar then the circuit is probably working. However, if they are different then this indicates there may be a problem.

Not surprisingly, the multimeter can be used to check digital electronics circuits. Remember though, not all digital systems use 0V and 5V for logic-0 and logic-1. Remember also that, even in systems that do use these two voltages, the measured values will never be exactly 0V and 5V.

To make the repair technician's life easier, specialised digital test equipment has been developed including: the logic probe; logic pulser; logic clip; and the logic comparator.

The logic probe

The logic probe is a device that, in many respects, is similar to a multimeter because it can be used to measure the voltage at a node in the circuit. But, instead of displaying the voltage's value, the logic probe uses two light-emitting diodes (LEDs) to simply indicate whether the voltage is within range for a logic-0 or a logic-1. An example of a logic probe is shown in Figure 1 below.

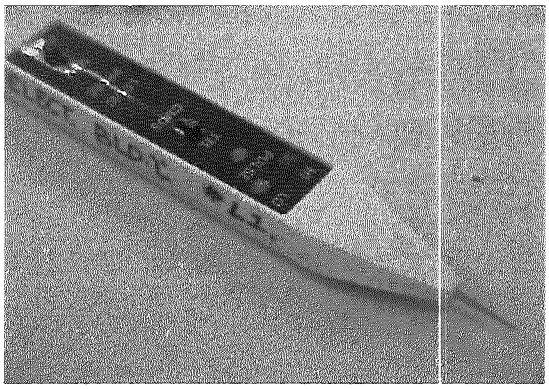


Figure 1 *An example of a logic probe*

Unlike a digital multimeter, the logic probe is not battery operated. It must draw its power from an external power supply and, to work accurately, this power supply must be same as the supply that powers the digital circuit being tested.

The logic pulser

The logic pulser is designed to produce bursts of current that can be injected into a digital circuit to momentarily over-ride the existing logic level at that point in the circuit. With a logic pulser connected to the input of a digital circuit and a logic probe connected to the output, the circuit can be tested to see if it responds appropriately to changes in logic levels. An example of a logic pulser is shown in Figure 2 below.

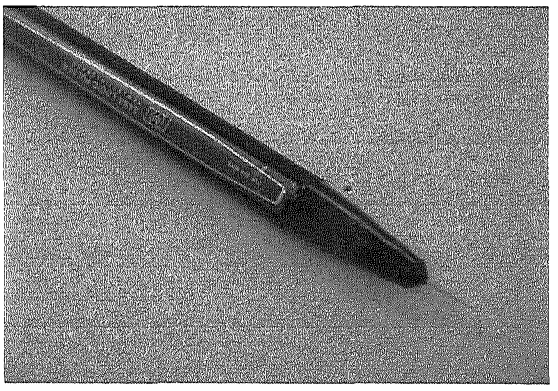


Figure 2 *An example of a logic pulser*

Logic clips

The logic clip can be thought of as eighteen logic probes in one unit. The device is designed to sit on top of 14-pin and 16-pin DIL IC packages and has one small red LED for each pin of the chip. These LEDs are used to show the logic level on all pins of the chip simultaneously. When the LED is off, the voltage on the pin is a logic-0. When the LED is on, the voltage on the pin is a logic-1. An example of a logic clip is shown in Figure 3 below and, despite its small size, they are expensive (over $300 each!).

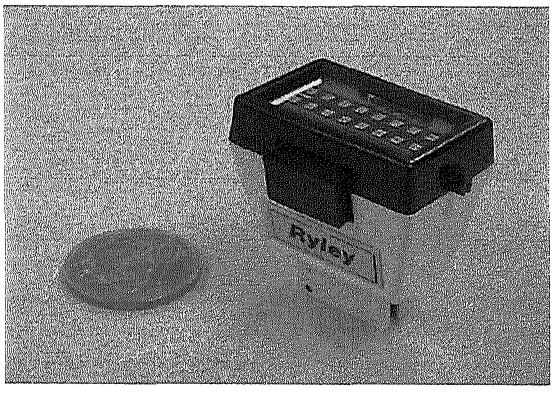


Figure 3 *An example of a logic clip*

Logic comparator

The logic comparator measures the logic levels on all of the pins of an IC in circuit (like a logic clip does) and at the same time measures the logic levels on all of the pins of a known functional version of the same type of chip (called the reference IC) that is connected in parallel with the IC under test. It then compares the logic levels on the pins of the two chips and if any of them are different (which of course they shouldn't be) an LED for that pin is turned on.

The reference IC is soldered onto a circuit board that is inserted into the comparator via a draw on the end of the unit. It is then connected in parallel with the IC under test via a lead and IC clip. An example of a logic comparator is shown in Figure 4 below.

Figure 4 *An example of a logic comparator*

Appendix 2

Circuits that implement RS flip flops with synchronous R & S inputs

Although you don't need to be able to draw the circuit of a gate-controlled RS flip flops or explain their operation, the circuits are not very difficult. So, for those that are interested, gated and clocked RS flip flops are explained here.

Gated RS flip flops

The basic circuit of a gated RS FF and the principle of operation will be demonstrated using a NOR RS FF. Figure 1 below shows the circuit of a NOR RS FF with a gate input.

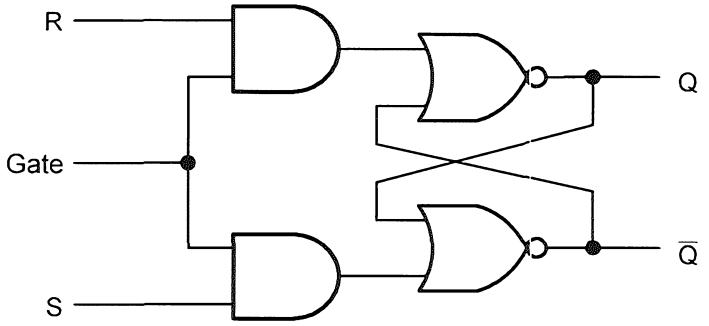


Figure 1 *The circuit of a NOR RS FF with a positive level-sensitive gate input*

Notice that the circuit is based on the original ungated NOR RS FF but with some additional circuitry for the gate control. To help with the explanation of the circuit's operation, the inputs to the original ungated NOR RS FF will be denoted by S' and R' (see Figure 2 on the next page).

Figure 2 below shows that when the gate input is logic-0, the outputs of the two AND gates are logic-0 as well. This is true regardless of the logic levels on the *S* and *R* inputs. With a logic-0 on the output of both AND gates, the *S'* and *R'* inputs to the ungated flip flop are logic-0 and so no change of logic level can occur on the flip flop's outputs. In other words, the circuit can't read its inputs even if they're being asserted.

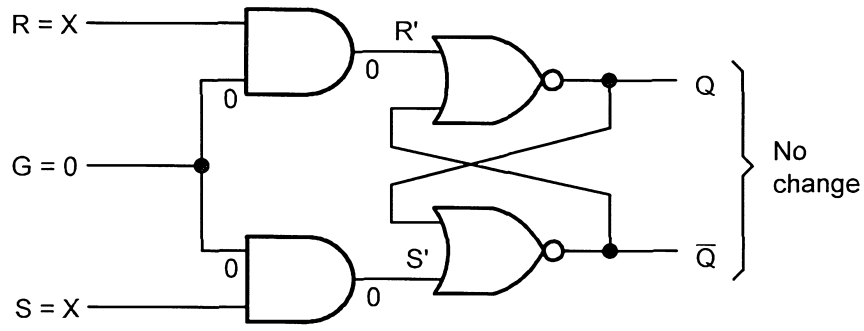


Figure 2 *The circuit conditions when the gate input is logic-0*

When the gate input is a logic-1, the output of the two AND gates is the same as their other input (that is, the input not connected to the gate control input of the circuit). So, as Figure 3 below shows, when the gate and *S* inputs are asserted, the Q-output gets set.

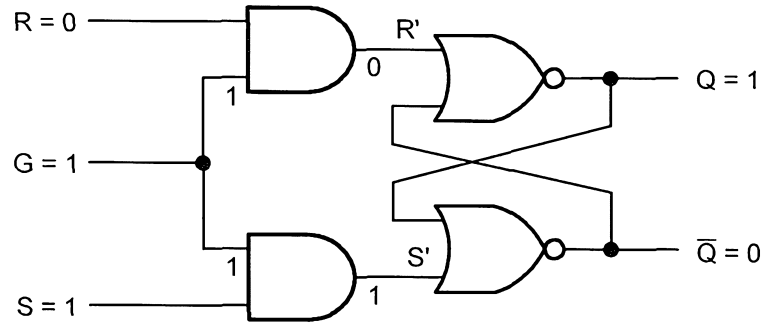


Figure 3 *The circuit conditions when the gate and set inputs are logic-1*

And Figure 4 below shows that, when the gate and *R* inputs are asserted, the Q-output gets reset.

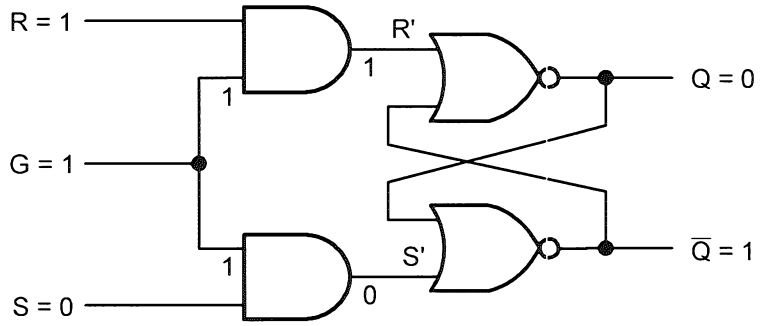


Figure 4 *The circuit conditions when the gate and reset inputs are logic-1*

From this explanation of the circuit's operation, we can see that the gate needs a logic-1 on it to allow the RS FF to read its inputs. So, the gate input is active high. When the operation of this sequential logic NOR RS FF is summarised in a truth table we get Table 1 below.

Table 1

$\overline{G}$	S	R	Q	$\overline{Q}$	Description
0	×	×	Q	$\overline{Q}$	Inputs not read
1	0	0	Q	$\overline{Q}$	No change
1	0	1	0	1	Q is reset
1	1	0	1	0	Q is set
1	1	1	0	0	Disallowed

To make the circuit's gate active low (that is, have the circuit controlled by a logic-0 on the gate input instead of a logic-1) a NOT gate can be added to the circuit as shown in Figure 5 below.

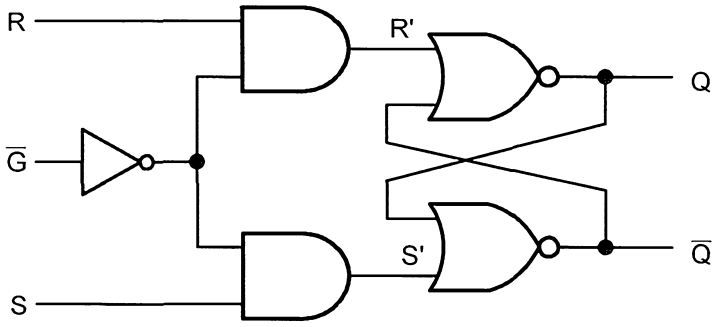


Figure 5 *The circuit of a gated NOR RS FF with an active low gate input*

When we summarise the operation of this sequential logic NOR RS FF in a truth table we get Table 2 below. Notice that, to emphasise that the gate is active low, the letter G in the truth table has a bar across it.

Table 2

$\overline{G}$	S	R	Q	$\overline{Q}$	Description
1	×	×	Q	$\overline{Q}$	Inputs not read
0	0	0	Q	$\overline{Q}$	No change
0	0	1	0	1	Q is reset
0	1	0	1	0	Q is set
0	1	1	0	0	Disallowed

Clocked RS flip flops

The basic circuit of a clocked RS FF and the principle of operation will be demonstrated using a NAND RS FF. Figure 6 below shows the circuit of a positive edge-triggered NAND RS FF.

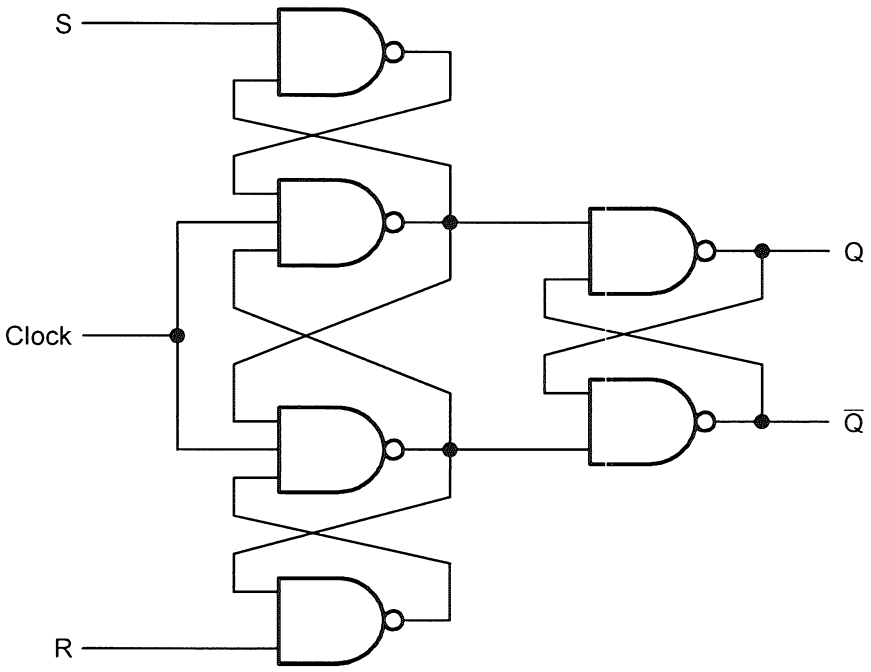


Figure 6 *The circuit of a positive edge-triggered NAND RS FF*

It would take many pages to explain the operation of this circuit showing all the logic levels around the circuit every time they changed so this approach wont be taken. Instead, consider the circuit as it is shown in Figure 7.

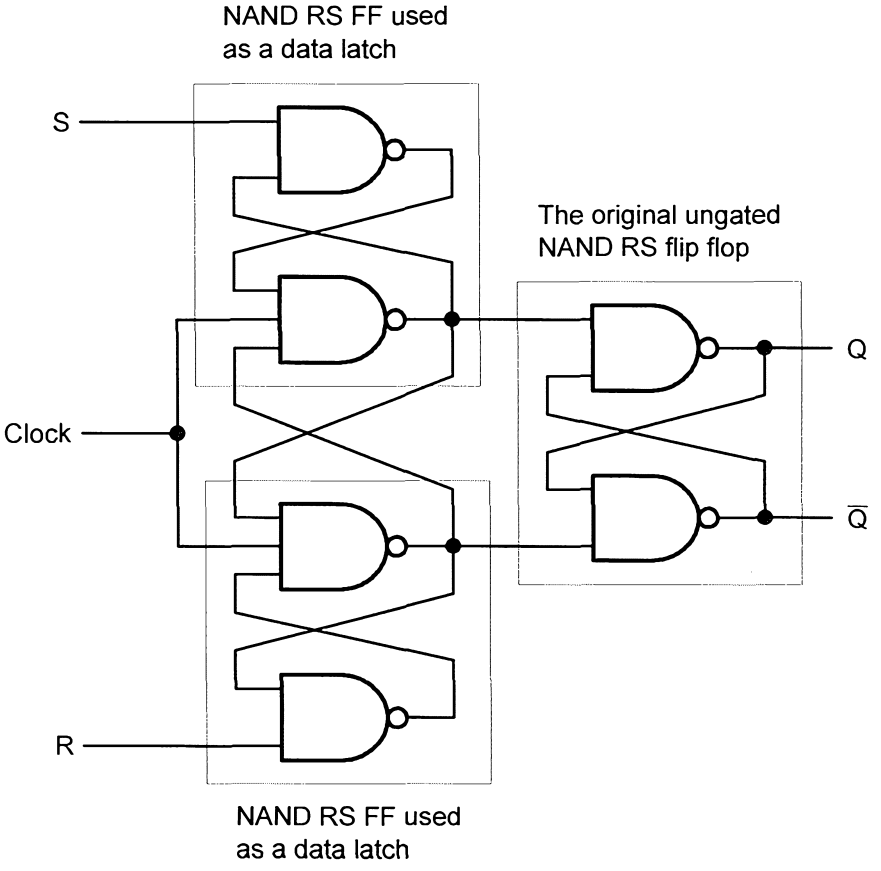


Figure 7

As Figure 7 shows, the clocked NAND RS FF consists of the original unclocked NAND RS FF plus two extra NAND RS flip flops used as data latches.

When the clock input changes from a logic-0 to a logic-1 it activates the two data latches and the logic level on their other inputs (*S* and *R*) is transferred to the latches' output. This data feeds the input of the original unclocked NAND RS FF affecting its outputs according to its truth table.

But, once this happens, as long as the clock remains a logic-1, the outputs of the two latches wont change again even if the data on the *S* and *R* inputs changes. So, in effect, the *S* and *R* inputs are read on the positive edge of the clock signal.

To allow the circuit to read its inputs again, the circuit's clock must be returned to a logic-0. This makes the data latches' outputs logic-1 which, at this stage, makes the original unclocked NAND RS FF's inputs logic-1. Recall that this condition is the "no change" input for a NAND RS FF so nothing happens yet. However, when the clock input changes from logic-0 to logic-1 again, the data on the *S* and *R* inputs is transferred to the original NAND RS FF's inputs again affecting its output accordingly.

Copyright: Not to be reproduced without permission

Appendix 3

Sample digital IC data (The 74ALS00)

The data herein is National Semiconductor's copyrighted material.

Copyright © 2003 National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, CA 95051, U.S.A

All rights reserved.

The data has been reprinted here with permission from National Semiconductor Corporation (http://www.national.com).

National Semiconductor

August 1995

DM74ALS00A
Quad 2-Input NAND Gate

General Description

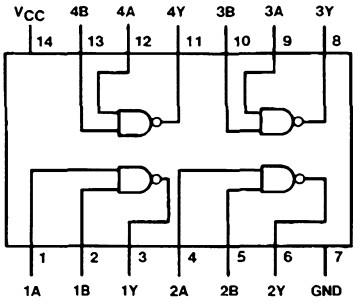
This device contains four independent gates, each of which performs the logic NAND function.

Features

- Switching specifications at 50 pF
- Switching specifications guaranteed over full temperature and V_{CC} range
- Advanced oxide-isolated, ion-implanted Schottky TTL process
- Functionally and pin for pin compatible with Schottky and low power Schottky TTL counterpart
- Improved AC performance over Schottky and low power Schottky counterparts

Connection Diagram

Dual-In-Line Package



TL/F/6270–1

Order Number DM74ALS00M, DM74ALS00SJ or DM74ALS00N
See NS Package Number M14A, M14D or N14A

Function Table

$Y = \overline{AB}$

Inputs		Output
A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

H = High Logic Level

L = Low Logic Level

©1995 National Semiconductor Corporation

RRD-B30M115/Printed in U. S. A.

Absolute Maximum Ratings

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM74ALS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C
Typical θ_{JA}	
N Package	86.5°C/W
M Package	116.0°C/W

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	DM74ALS00A			Units
		Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	V
V_{IH}	High Level Input Voltage	2			V
V_{IL}	Low Level Input Voltage			0.8	V
I_{OH}	High Level Output Current			−0.4	mA
I_{OL}	Low Level Output Current			8	mA
T_A	Free Air Operating Temperature	0		70	°C

Electrical Characteristics

over recommended operating free air temperature range. All typical values are measured at V_{CC} = 5V, T_A = 25°C.

Symbol	Parameter	Conditions		Min	Typ	Max	Units
V_{IK}	Input Clamp Voltage	V_{CC} = 4.5V, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	I_{OH} = −0.4 mA V_{CC} = 4.5V to 5.5V		V_{CC} − 2			V
V_{OL}	Low Level Output Voltage	V_{CC} = 4.5V	74ALS I_{OL} = 8 mA		0.35	0.5	V
I_I	Input Current at Max Input Voltage	V_{CC} = 5.5V, V_{IH} = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = 5.5V, V_{IH} = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = 5.5V, V_{IL} = 0.4V				−0.1	mA
I_O	Output Drive Current	V_{CC} = 5.5V	V_O = 2.25V	−30		−112	mA
I_{CC}	Supply Current	V_{CC} = 5.5V	Outputs High		0.43	0.85	mA
			Outputs Low		1.62	3	mA

Switching Characteristics over recommended operating free air temperature range (Note 1)

Symbol	Parameter	Conditions	DM74ALS00A		Units
			Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	V_{CC} = 4.5V to 5.5V R_L = 500Ω C_L = 50 pF	3	11	ns
t_{PHL}	Propagation Delay Time High to Low Level Output		2	8	ns

Note 1: See Section 5 for test waveforms and output load.

Answers to review questions

Section 1

1. The signal in Figure 1 is not a digital signal.

2. Because the signal changes size in a continuous action.

3. Because the signals change size in a stepped action.

   (It might be tempting to argue here that the signal in Figure 1 is digital and that its changing in steps but they're just too small to see. This could be confirmed by magnifying the signal - If we never see steps then it's truly analog but if we do eventually see steps then its digital.)

4. Because digital circuits (such as logic gates) are designed to work with pairs of voltages so Figure 2 has too many voltage levels.

5. Clock radios
   Televisions
   PVRs
   Blu-ray players
   Hi-fi systems (especially the CD player but other components will too)
   Computers
   Modems
   Microwave ovens
   Washing machines and most other white goods (because they're mostly microcontroller controlled nowadays and these are digital devices)

6. Because digital signals are inherently less affected by noise than analog signals.

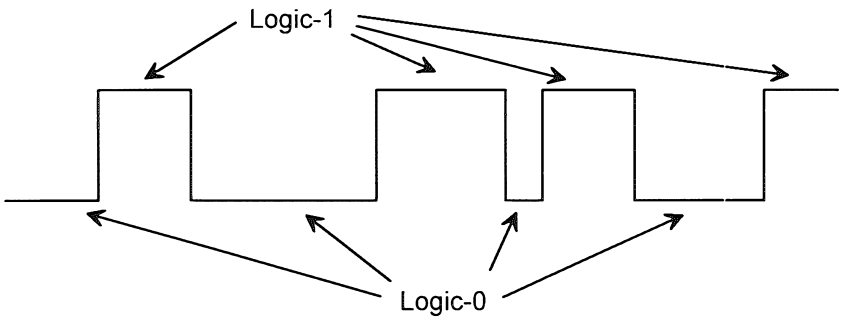
7. It is possible to electrically store information in digital form that is impossible (or very complex) to store in analog form (eg text).

   Greater storage capability of information, sound, pictures and video when they're converted to digital.

   Information, sound, pictures and video can be manipulated in ways that are impossible (or very difficult) when they are in analog form.

Section 1 (continued)

8.



9. Low, false, open and off.

10. High, true, closed and on.

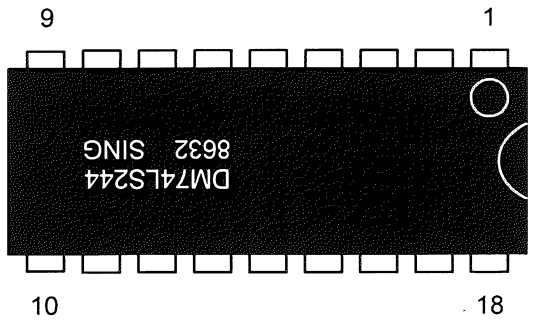
11. Logic-0 is ideally represented by 0V (though voltages near 0V are acceptable)
Logic-1 is ideally represented by 5V (though voltages near 5V are acceptable)

12. DIL (which stands for dual in-line) and SOIC (which stands for small outline integrated circuit).

13. PLCC (which stands for plastic, leadless chip carrier) and LCCC (which stands for leadless ceramic chip carrier).

14. PGA (which stands for pin grid array).

15.



Note: The notch and the dimple indicate pin-1. When the chip is oriented so that the notch and or dimple are on the left hand side pin-1 is directly below them.

16. Although both the DMM and the logic probe measure voltage, the logic probe automatically converts the measurement to a logic-0 or a logic-1 and displays this accordingly. This saves the technician having to convert the voltages themselves.

Section 1 (continued)

17. After 0100 comes 0101
    After 1011 comes 1100
    After 0111 comes 1000

18. electrons between two surfaces.

19. triboelectric charging.

20. As two surfaces rub together, they constantly touch and separate. In the process, electrons jump off one surface and accumulate on the other creating an imbalance of electrons between the two surfaces.

21. equalises.

22. Most digital ICs are designed to withstand only relatively small voltages. However, the voltages involved in ESD events can be much larger.

23. The wrist-strap keeps the wearer at or near ground potential so there is no electrostatic charge to discharge via the component or circuit board.

    The wrist-strap is connected to earth via a 1MΩ resistor. So, if a charged component or circuit board discharges via the wearer, the currents may be low enough to prevent damage.

24. Work on an anti-static surface.
    Use test equipment and tools that are grounded.
    Watch out for the ESD association awareness symbols.
    Treat all components as though they are especially ESD sensitive.

25. True.

26. False. In fact, most of them are below 3kV which is why you can't feel them.

27. False. It's probably closer to say that most electronics components can be damaged by ESD events up to 4kV.

28. False. The susceptibility to damage from ESD for components doesn't change. However, on warm dry days, ESD events involve much higher voltages which increases the chances of damage for any components that experience an ESD event.

29. True. The susceptibility to damage from ESD for components doesn't change. However, when mounted on a circuit board, the chances of a component experiencing an ESD event increases.

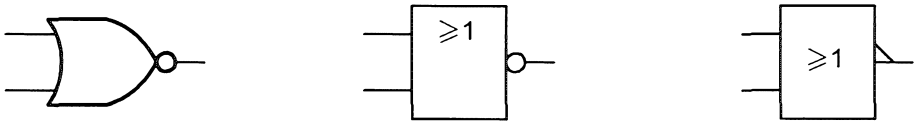
30. True.

31. False. Wearing a wrist-strip only minimises ESD events.

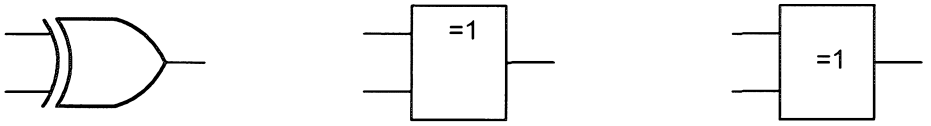
 Copyright: Not to be reproduced without permission

Section 2

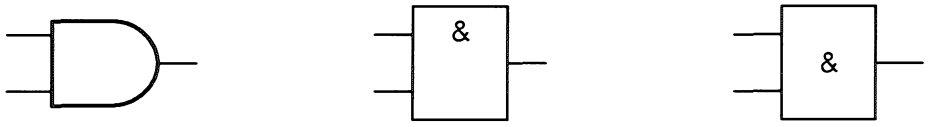
1. You should have circled the gates shown below.



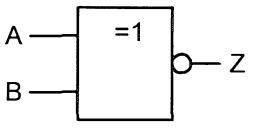
2. You should have circled the gates shown below.



3. You should have circled the gates shown below.



4.



Section 2 (continued)

5.

a) The output of a NAND gate will be **false** when both input-A and input-B are true. For all other input conditions, the output will be **true**.

b) The output of an OR gate will be true if either input-A or input-B is true. If input-A and input-B are false, the output will be false.

c) The output of an XOR gate will be **false** when input-A and input-B are the same. For all other input conditions, the output will be **true**.

d) The output of the **NOT** gate will be true when the input is false. If the input is true, the output will be false.

6. The first three answers are all correct. There's no such thing as an OT gate.

7. OR
   AND
   XOR

8. AND
   NOR
   XNOR

9. NOR
   NAND
   XOR

10. AND

11. NAND

12. OR

13. NOR

14. OR

15. NOR

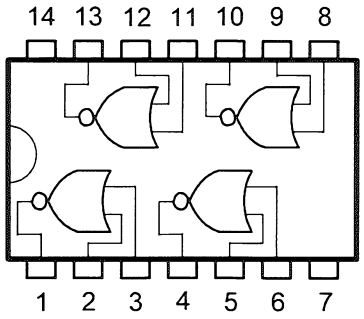
16. AND

17. NAND

Copyright: Not to be reproduced without permission

Section 2 (continued)

18.



19. 2, 3, 5, 6, 8, 9, 11, 12

20. 1, 4, 10, 13

21. These are the power supply connections (usually, pin-14 is the +5V input and pin-7 is the ground input).

Section 3

1. To implement one of the seven logic functions using the other gates.
   To implement truth tables not provided for by the seven logic gates.
   To implement digital functions not available with single gates (such as adders).
   To control electrical, mechanical and electromechanical process.

2.

B	A	Z
0	0	0
0	1	1
1	0	1
1	1	0

3. An XOR gate

4.

B	A	Z
0	0	0
0	1	1
1	0	1
1	1	1

5. A 2-input OR gate

6.

B	A	Z
0	0	0
0	1	0
1	0	0
1	1	1

7. A 2-input AND gate

 Copyright: Not to be reproduced without permission

Section 3 (continued)

8.

B	A	Z
0	0	0
0	1	0
1	0	1
1	1	0

9. There isn't one.

10.

B	A	X	Y	Z
0	0	1	1	0
0	1	0	0	1
1	0	1	1	0
1	1	0	1	0

11.

B	A	X	Y	Z
0	0	1	1	0
0	1	0	1	0
1	0	1	0	1
1	1	0	1	0

12.

C	B	A	X	Y	Z
0	0	0	0	0	1
0	0	1	1	1	0
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	0	1	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	1	0

13. A 3-input NOR gate

Section 3 (continued)

14. Isolate the output of the AND gate (gate 4) from the input of the OR gate (gate 5) and test the AND gate's output. If the logic level is still incorrect then the AND gate is faulty. If the logic level is correct then the OR gate is faulty.

15.

C	B	A	X	Y	Z
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	0	0	0

16. Gate 5

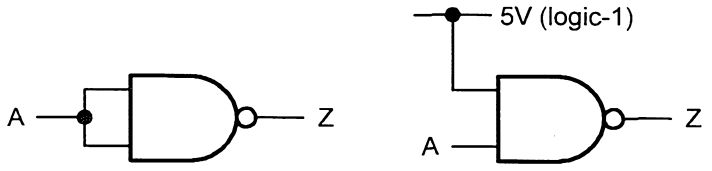
17. Gates 5 or 6

18. Gates 2 or 4

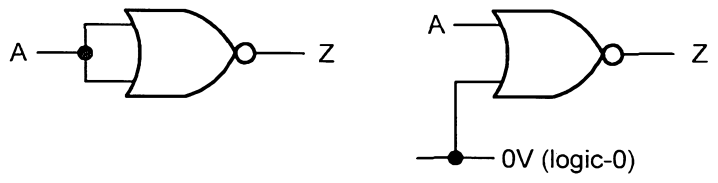
19. Gates 3 or 5

20. Gates 2 or 4 or 5 or 6

21.

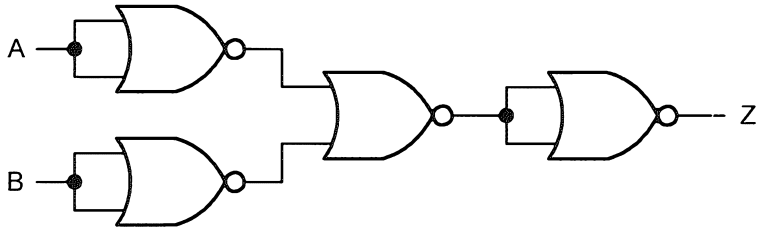


22.



Section 3 (continued)

23.



Section 4

1. (a) $Z = A \cdot B$ or $Z = AB$
   (b) $Z = \overline{A}$

2. (a) NAND
   (b) OR
   (c) XOR

3. (a) $Z = \overline{\overline{AB}}$ Two NOT signs across a variable or group of variables cancel which gives:
   $Z = AB$

   (b) $Z = \overline{A} + \overline{B}$ When you make the line, change the sign so this gives:
   $Z = \overline{A \cdot B}$

   (c) $Z = \overline{\overline{A} + B}$ When you break the line, change the sign giving:
   $Z = \overline{\overline{A} \cdot \overline{B}}$ And double bars cancel giving:
   $Z = A \cdot \overline{B}$

   (d) $Z = \overline{A\,\overline{\overline{B}}}$
   $Z = \overline{A} + \overline{\overline{B}}$
   $Z = \overline{A} + B$

   (e) $Z = \overline{A}\,\overline{\overline{B}}$
   $Z = \overline{A}B$ or $Z = \overline{A + \overline{\overline{B}}}$

   (f) $Z = \overline{\overline{\overline{AB}}C}$
   $Z = \overline{\overline{AB}} + \overline{C}$
   $Z = AB + \overline{C}$

   (g) $Z = \overline{AB} + \overline{C}$
   $Z = \overline{ABC}$ or $Z = \overline{A} + \overline{B} + \overline{C}$

Section 4 (continued)

(h) $Z = \overline{A} + \overline{B + C}$

$Z = \overline{AB + C}$ or $Z = \overline{A} + \overline{B}\,\overline{C}$

(i) $Z = \overline{\overline{A} + \overline{B}C}$

$Z = \overline{\overline{A}} \cdot (\overline{\overline{B}} + \overline{C})$

$Z = A \cdot (B + \overline{C})$

(j) $Z = \overline{\overline{A}B\overline{\overline{C}}}$

$Z = \overline{\overline{A}} + \overline{B} + \overline{\overline{C}}$

$Z = A + \overline{B} + C$

4. Figure 1: $Z = \overline{\overline{A}B}$

Figure 2: $Z = \overline{\overline{A + B + C}}$

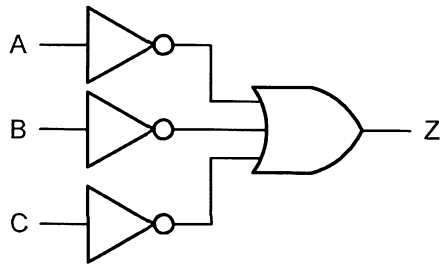
Figure 3: $Z = \overline{AB\overline{\overline{C}}}$

Figure 4: $Z = \overline{A}B\overline{C} + \overline{B}C$

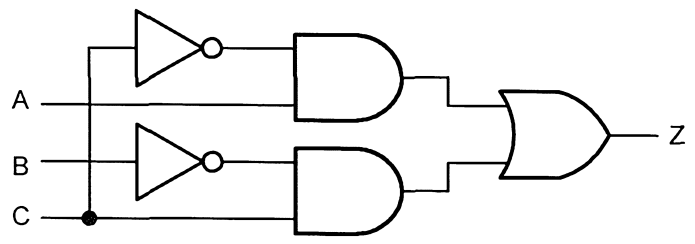
Figure 5: $Z = \overline{A + C} + \overline{B + C} + BC$

Figure 6: $Z = AB\overline{C} + \overline{A}\,\overline{B}C$

5.



6.



 Copyright: Not to be reproduced without permission

Section 4 (continued)

7.

a)

B	A	Z
0	0	0
0	1	1
1	0	0
1	1	1

b)

B	A	Z
0	0	1
0	1	1
1	0	1
1	1	0

c)

C	B	A	Z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

d)

C	B	A	Z
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

8. $Z = \overline{A}\overline{B} + AB$

B	A	Z
0	0	1
0	1	0
1	0	0
1	1	1

9. $Z = \overline{A}\overline{C}B + \overline{B}C$

C	B	A	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Section 5

1.
   a) Commercial grade low power schottky TTL
   b) Mil-spec standard TTL
   c) Commercial grade, 4000 series CMOS
   d) Mil-spec advanced low power schottky TTL
   e) Commercial grade 74 series CMOS
   f) Commercial grade advanced schottky TTL
   g) Mil-spec advanced CMOS

2. Two NAND gates each having four inputs.

3. Six NOT gates

4. 4000 series CMOS operates over a much wider range of supply voltages.

5. TTL is less susceptible to damage due to ESD than 4000 series CMOS.

6.

a)	74LS32	4.75V to 5.25V (based on 5V ± 5% for commercial grade TTL)
b)	54193	4.5V to 5.5V (based on 5V ± 10% for military grade TTL)
c)	4011	3V to 18V
d)	74HC40	2V to 6V
e)	74HCT10	4.5V to 5.5V

7. 54 series can operate on a wider range of supply voltages than 74 series.
   54 series can operate over a wider range of temperatures than 74 series.

8. propagation delay.

9. fanout.

10. Noise margin is the term for the maximum allowable amount of electrical noise on the inputs to a logic device. [Once exceeded, the voltages on the gates input fall into its transition zone.]

11. You would expect to see this nstead of this

12. 7400, 74AS121, 74ALS04, 54AS74, 74LS193

13. 74C32, 74ACT86, 4011, 74HC08

 Copyright: Not to be reproduced without permission

Section 5 (continued)

14.

Device	V_{OH}	V_{OL}	V_{IH}	V_{IL}
74ALS series TTL	2.5V	0.4V	2.0V	0.8V
4000 series CMOS	V_{DD}	0V	$0.7V_{DD}$	$0.3V_{DD}$

15. Propagation delay, the maximum clock speed and the fanout.

16. Power dissipation.

17. None of them (you can't include the 4000 series of devices because it's CMOS, not TTL).

18. 4×8mW = 32mW [Note: the power rating is a per gate figure and there are four gates inside.]

19. Power dissipation and fanout.

20. Propagation delay and maximum clock speed.

21. The specifications for propagation delay, maximum clock speed and power dissipation for the 74ALS sub-family are all better than the same specifications for the 74LS sub-family.

    [Note: The IC's function number has nothing to do with this question - The 74ALS139 and the 74LS139 do the same thing.]

22. The specifications for propagation delay, power dissipation, maximum clock speed and fanout for the 74AS sub-family are all better than the same specifications for the original 74 TTL devices.

23. Because the 74AS00 has a different function to the 4000. In which case, the performance specifications are irrelevant.

24. 74ALS73

25. 2.7V to 5V [Note: the IC's function number has nothing to do with this question.]

Section 5 (continued)

26. 0V to 0.4V

27. 2V to 5V

28. 0V to 0.8V

29. Greater than 0.8V but less than 2V.

30. A maximum of 50µA.

31. A maximum of 0.4mA.

32. Unit load

33. A maximum of 0.4mA. [Note: The source current is the I_{OH}.]

34. A maximum of 20mA. [Note: The sink current is the I_{OL}.]

35. This must be calculated for both logic-0 and logic-1 and the worst figure chosen:

    $V_{OH} - V_{IH} = 2.7 - 2.0 = 0.7V$

    $V_{IL} - V_{OL} = 0.8 - 0.5 = 0.3V$

    So the noise margin for 74S series TTL is 300mV because you must take the worse of the two figures.

36. $V_{OH} - V_{IH} = 2.5 - 2.0 = 0.5V$

    $V_{IL} - V_{OL} = 0.8 - 0.5 = 0.3V$

    So the noise margin for 74AS series TTL is also 300mV

37. The maximum I_{OL} specification of CMOS is below the I_{IL} of the TTL inputs.

38. A CMOS buffer

39. The minimum V_{OH} specification of TTL is below the minimum V_{IH} of CMOS.

40. A pull-up resistor

41. 74HCT

 Copyright: Not to be reproduced without permission

Section 6

1.
   a) 6
   b) 4
   c) 11
   d) 13
   e) 9
   f) 17
   g) 102
   h) 159

2.
   a) 1000
   b) 1110

3.
   a) 00001000
   b) 00001110
   c) 00010010
   d) 00101010
   e) 01000110
   f) 01100101
   g) 10000000
   h) 11111000

4.
   a) $8
   b) $E
   c) $03
   d) $0C
   e) $11
   f) $66
   g) $9F
   h) $D1A

5.
   a) 0010
   b) 1010

6.
   a) 00000101
   b) 00001011
   c) 00110110
   d) 01111111
   e) 10111101
   f) 11111111

Section 6 (continued)

7.

a) 011011100010
b) 101010111100

8.

a) 0001111111111111
b) 0010000000000000

9.

a) 21
b) 44
c) 146
d) 179
e) 193
f) 431
g) 1024
h) 2072

10.

a) $1D
b) $32
c) $6B
d) $FF
e) $100
f) $148
g) $19B
h) $4BF

 Copyright: Not to be reproduced without permission

Section 7

1. (a) 0111
   (b) 00010000
   (c) 00101001
   (d) 001101011000
   (e) 0001010001110110

2. (a) 3
   (b) 21
   (c) 98
   (d) 457
   (e) 1253

3. Because the first four digits of the number (starting from the LSB) are 1011 and this is not a valid code in BCD.

4. It's easier for people to convert between decimal and BCD; or
   It's easier to design circuits that convert decimal information into BCD and vice versa.

5. Circuitry that can perform mathematical operations on BCD numbers is much more complicated than circuitry that perform mathematical operations on true binary numbers.

   Note: It is common to called binary numbers "true" binary because BCD and ASCII codes (as well as others not mentioned in this module) are made up of zeros and ones and look like binary numbers.

6. Digital clocks
   The keypad for alarm systems
   Digital speedometers in cars
   Multimeters

7. (a) 0110001 or $31
   (b) 1110101 or $75
   (c) 0010111 or $17
   (d) 0111110 or $3E
   (e) 1111110 or $7E

8. (a) ACK
   (b) :
   (c) ESC
   (d) DEL
   (e) =

Section 7 (continued)

9. Because there are only seven digits. BCD codes are made up of multiples of four digits (that is, four digits, eight digits, twelve digits and so on).

10. Rendering the textual information of websites on the screens of mobile phones, computers, etc.

11. Two or more bits can change between successive binary numbers whereas only one bit changes between successive Gray code values.

12. Binary can generate transitory codes between the number changes confusing the digital electronics as to the shaft's position.

13. 101⇨**100**⇨110
    101⇨**111**⇨110

 Copyright: Not to be reproduced without permission

Section 8

1.

2. Limit the current through the LED to prevent it from being damaged.

3. $R = \frac{V_R}{I_{LED}}$

   $R = \frac{9V - 2.2V}{12mA}$

   $R = 567\Omega$

4. 560Ω

5. The circuit would probably still work but the LED would not glow a fraction less noticeably bright as the same circuit with a 560Ω resistor.

6. The circuit current would be too small to operate the LED.

7. Logic-1. [This is because a logic-0 is needed on the NOT gate's output.]

8. 160Ω

9. Clock radios, oven clocks.

Section 8 (continued)

10. Eight. One each for the segments and one for the decimal point.

11. b,c, f and g.

12. A common anode seven-segment display has the anode legs of all the LEDs inside the display connected together. A common cathode display has the cathode legs of all the LEDs connected together.

13. The supply rail.

14. Logic-0

15. Ground.

16. Logic-1

17. To convert BCD (or binary) input into a seven bit code that turns on or off the appropriate LEDs in a single seven-segment display.

18. Advantages include: Wider viewing angle
Can be seen in the dark

Disadvantages include: Hard to read in strong back-light situations
Not suitable for battery operated equipment (because of their current requirements)

19. Calculators, digital watches

20. A - polarised glass
B - segment electrodes
C - liquid crystal
D - backplane electrode
E - mirror

21. in the regions where the segment electrodes are active.

22. E - the mirror layer

23. connecting anti-phase squarewaves to the segment electrode and the backplane electrode.

24. 1

 Copyright: Not to be reproduced without permission

Section 8 (continued)

25. Because the backplane signal connects directly to the backplane electrode and one input of an XOR gate. If the segment control input is a logic-1, whenever the backplane signal is a logic-0 the output of the XOR gate is logic-1. Conversely, whenever the backplane signal is a logic-1, the output of the XOR gate is logic-0. That being the case, the output of the XOR gate is a squarewave that is in the opposite phase to the backplane signal.

26. Advantages include: Can display any kind of shape (like the Ohms symbol for example)
Can be read in strong back-light situations
Low current requirements make it ideal for battery operated equipment.

Disadvantages include: Narrow viewing angle
Can't be seen in the dark (without a backlight)

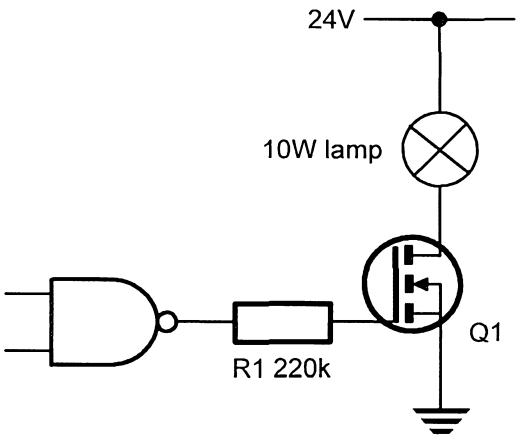
Section 9

1. To provide electrical isolation between the low voltage digital circuits and higher voltage/current loads (like AC motors).

   To translate between the operating voltage and current levels of digital circuits and the voltage and current requirements of electrical devices.

   To convert the non-voltage action and low-voltage action of most transducers into voltages that can be understood by the digital circuits.

2.



3. Logic-1

4. A collector resistor isn't needed because the lamp's own resistance self-limits the collector current.

5. Either a 1k2Ω or an 820Ω but preferably the latter.

6. Commutating diode or recirculating diode.

7. A commutating diode is necessary across inductive loads operating with DC voltages because, when they are switched off, inductors induce a back-emf which may exceed the rated V_{CE} of the transistor and destroy it. The diode effectively shorts this voltage preventing this from happening.

Section 9 (continued)

8. It is forward biased the moment the transistor is turned on. When this happens, the diode would act as a short across the load. With no other resistor in the collector arm, a large current would flow and either the diode or the transistor (or both) would be destroyed.

9. MOSFETs are voltage operated devices (that is, there's no gate current) so they can be driven from low power TTL devices and CMOS devices.

10. The disadvantages of relays are:

    a) they have moving parts and so are prone to mechanical wear and failure
    b) relays have a slow switching speed
    c) the contacts arc as they make and break causing contact wear (by burning them).

11. Switch bounce occurs on all switches as they make or break and contact. Their mechanical construction causes the contacts to make and break contact multiple times for the one switch throw.

12. On a computer keyboard, switch bounce might cause several letters to appear on the screen for only one press of a letter on the keyboard.

    If a switch is being used as an input to a counter then one activation of the switch might cause the counter to increase by much more than one.

13. Gates without input hysteresis have a V_{IL}, a transition zone and a V_{IH}. Gates with input hysteresis have a only V_{IL} and a V_{IH}.

    Moreover, the level that determines what an input voltage is read as changes. To be accepted as a logic-0 at first an input must drop below the gate's V_{IL}. However, once accepted as a logic-0 the input need only stay below the V_{IH} of the gate (which means it can exceed the gate's V_{IL}). The reverse is also true. Once accepted as a logic-1 the input voltage is allowed to drop below the V_{IH} and is still read as a logic-1 (until it drops below V_{IL}).

14. Debouncing is a method of ensuring that the multiple logic state changes that occur due to switch bounce are either eliminated (implemented in hardware) or ignored (implemented in software).

15. To provide electrical isolation between the two circuits (though, the circuit is very good at minimising problems associated with noise as well because noise causes light amplitude variations which are averaged out).

16. They are quicker.
    They have no moving parts so are not prone to mechanical wear.
    There are no contacts so there is no arcing which can burn the contacts.

Section 10

1. NOR RS flip flop.

2. *R* stands for reset and *S* stands for set.

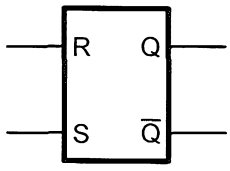
3. The Q and $\overline{Q}$ outputs should always be the opposite logic levels.

4. Sequential logic.

5.

S	R	Q	$\overline{Q}$	Description
0	0	Q	$\overline{Q}$	No change
0	1	0	1	Q is reset
1	0	1	0	Q is set
1	1	0	0	Disallowed

6.



7. asynchronous and active high.

8. NAND RS flip flop

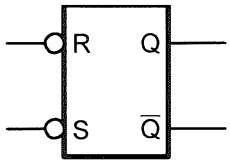
9.

$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$	Description
0	0	1	1	Disallowed
0	1	1	0	Q is set
1	0	0	1	Q is reset
1	1	Q	$\overline{Q}$	No change

 Copyright: Not to be reproduced without permission

Section 10 (continued)

10.

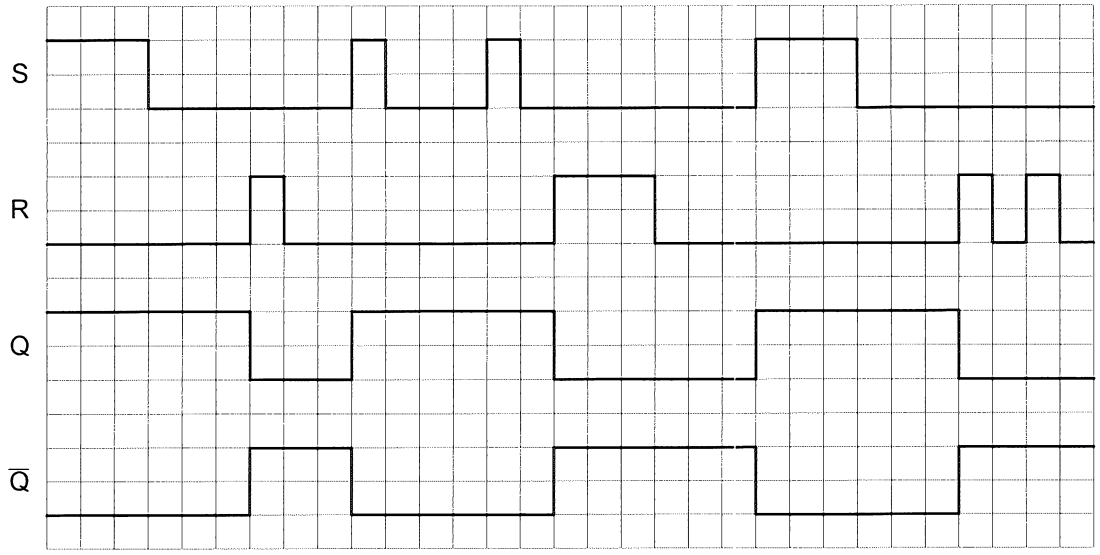


11. asynchronous and active low.

12. Because sometimes electrical events are very brief (like the press of a button) but a record of the event having taken place is needed (that is, the fact that the button was pressed is important and needs to be captured).

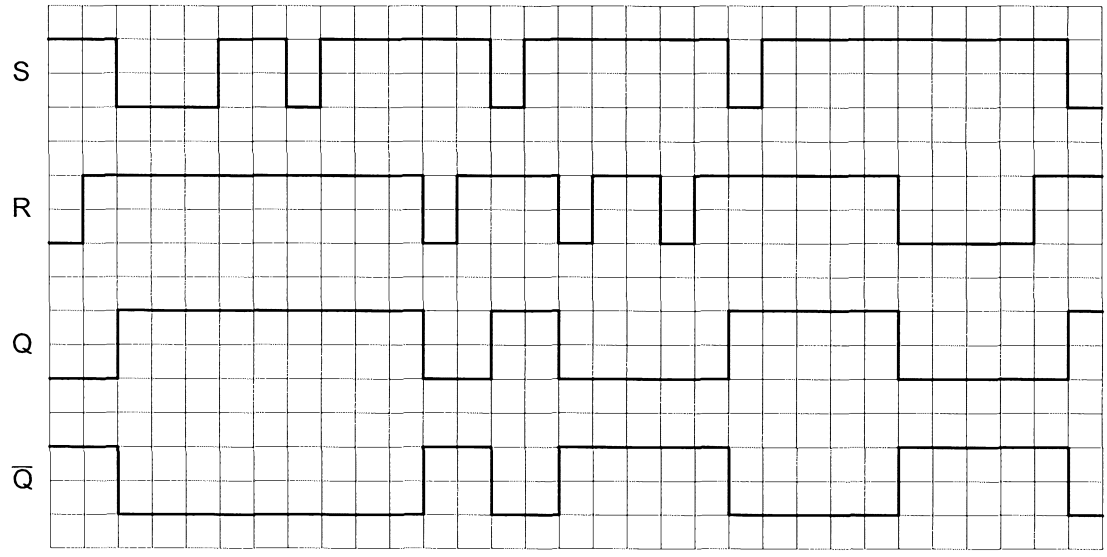
13. Call indicators for situations like elevators in buildings, nurses in hospitals, and flight attendants on planes.
The memory LED on a logic probe.
The pedestrian crossing button on a set of traffic lights.
The storage for a single bit of information in SRAM.

14.

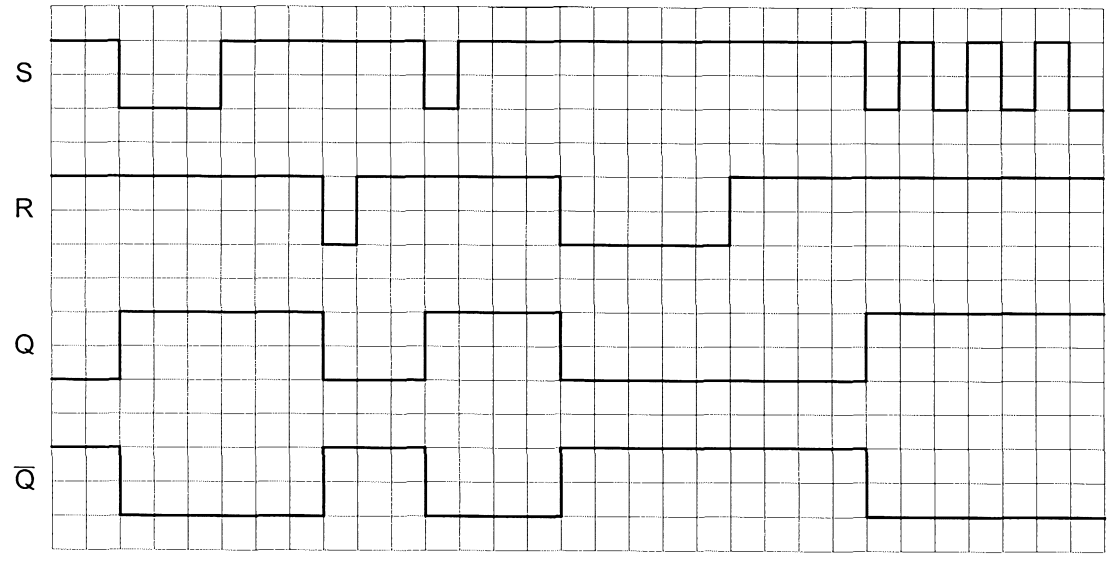


Section 10 (continued)

15.



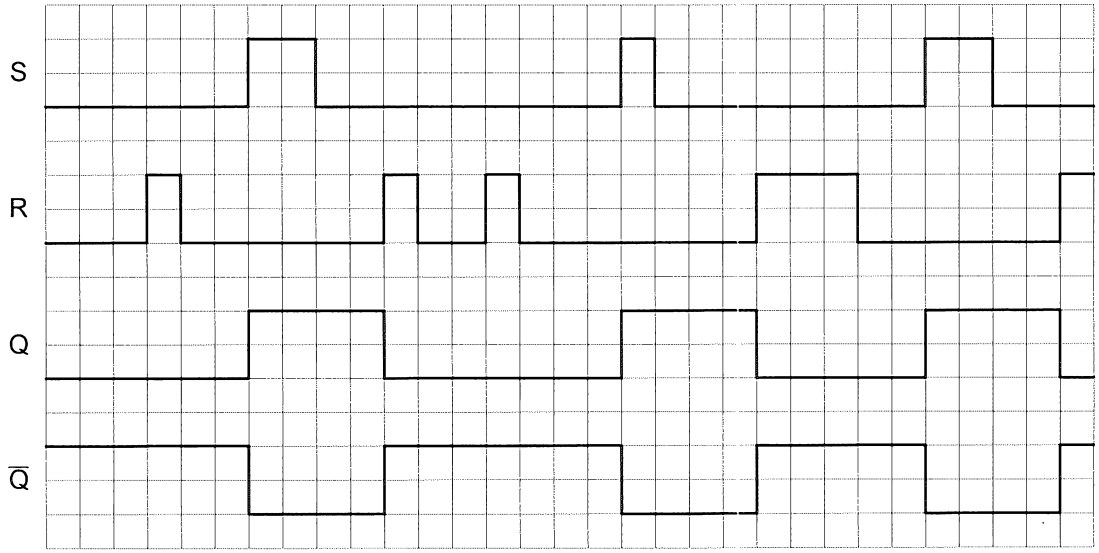
16.



 Copyright: Not to be reproduced without permission

Section 10 (continued)

17.



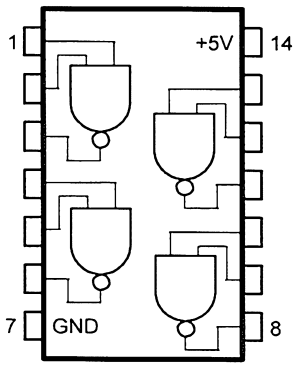
18. *R* and *S* only

19. *G*, *R* and *S*

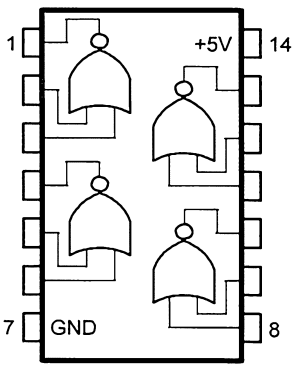
20. None of them

21. *G*, *R* and *S*

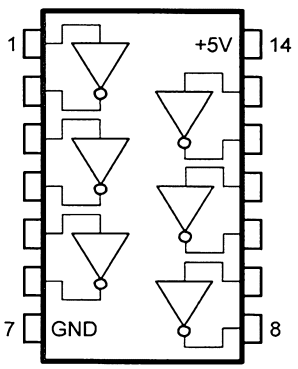
7400



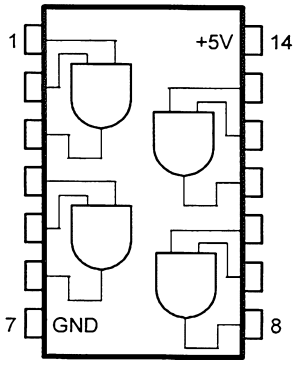
7402



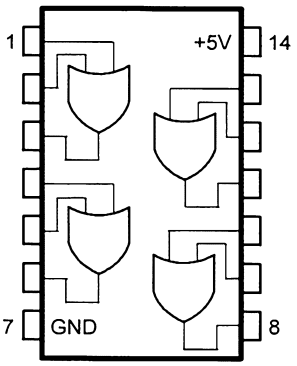
7404



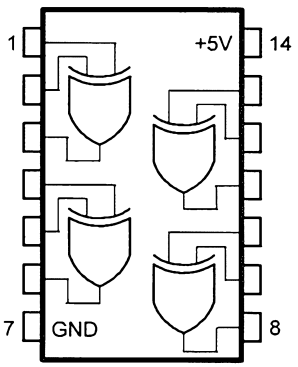
7408



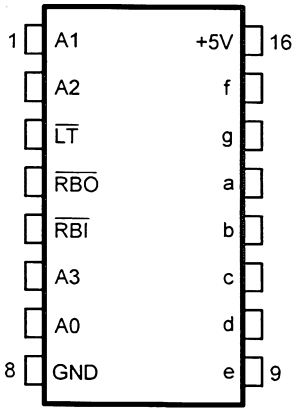
7432



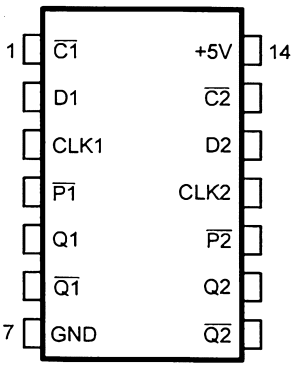
7486



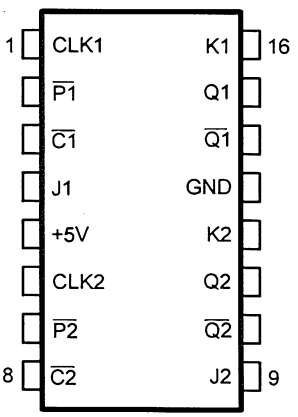
7447



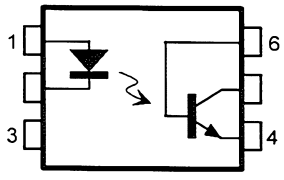
7474



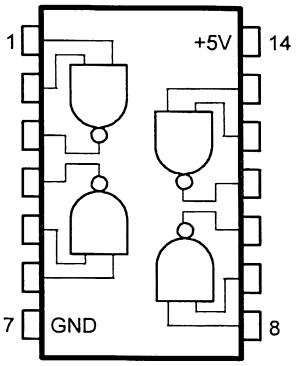
7476

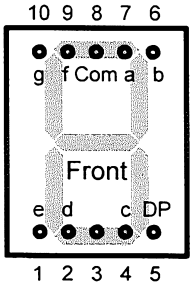


4N25



4011





LSD5365